



Real Business Learning...
Sponsored · mesaschool.co



Apply now

The useCallback hook in React | Sigma Web Development Course - Tutorial #118

80K views 1y ago Sigma Web Development Course - Web | ...more



CodeWithHarry 7.98M

Join



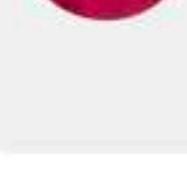
2.9K | 1.1K

Share

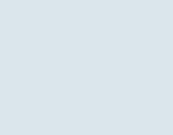
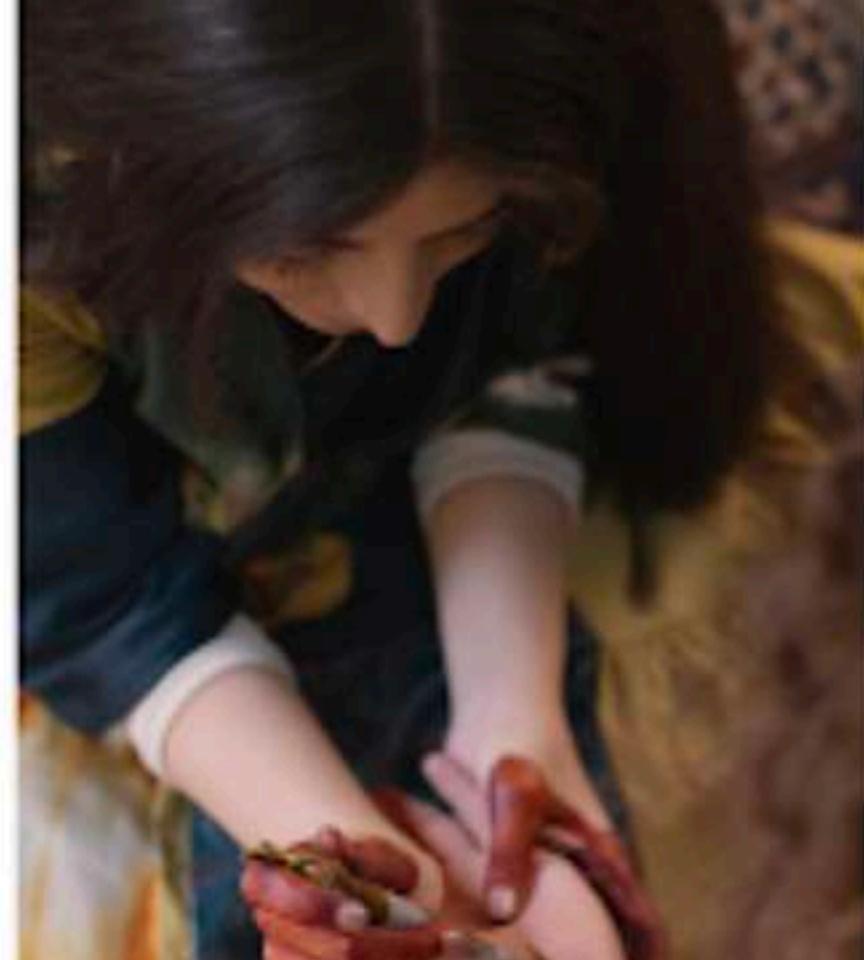
Remix

Download

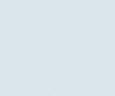
Comments 277



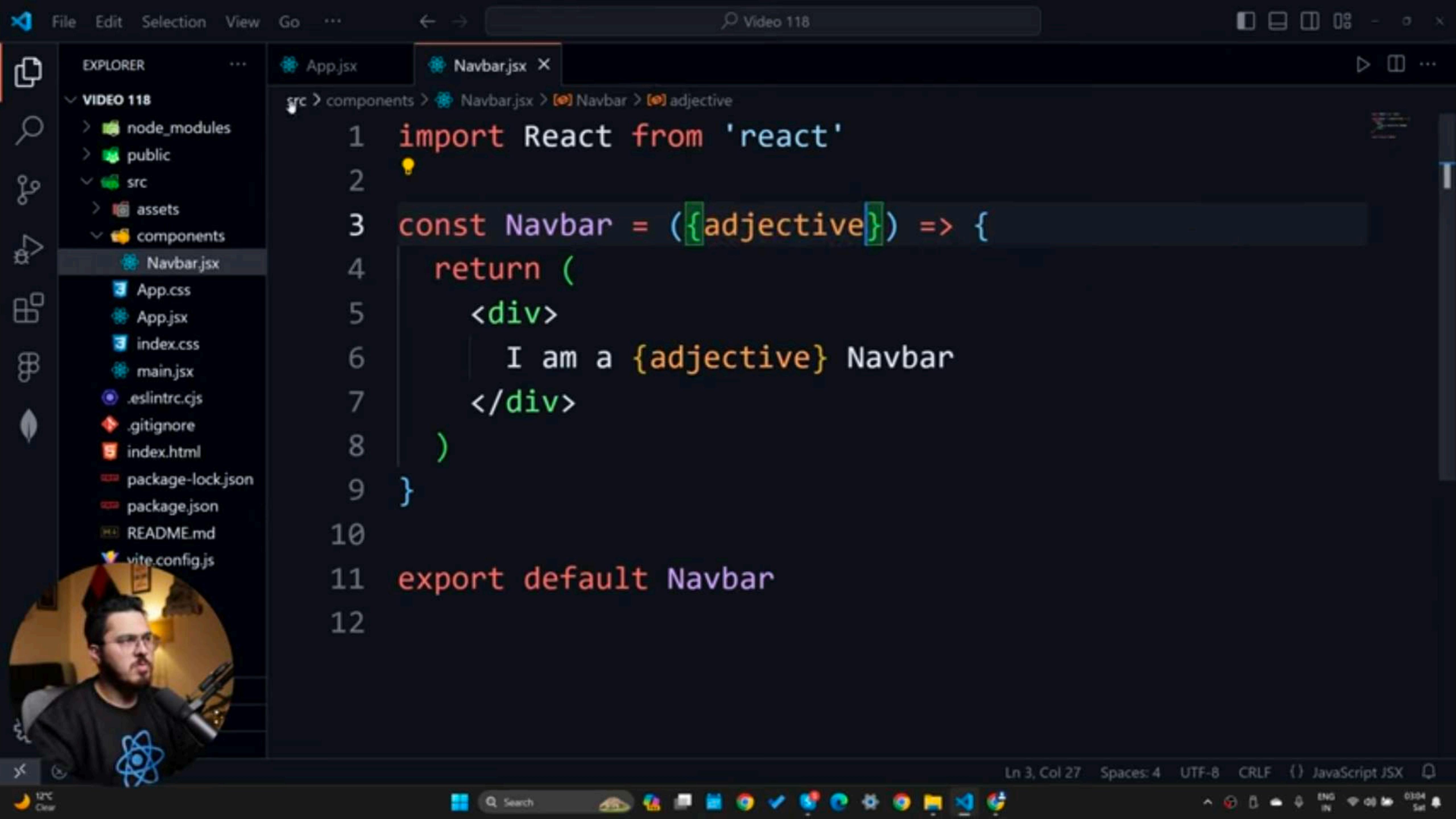
Bhai thumbnail me react ka logo to laga lo..

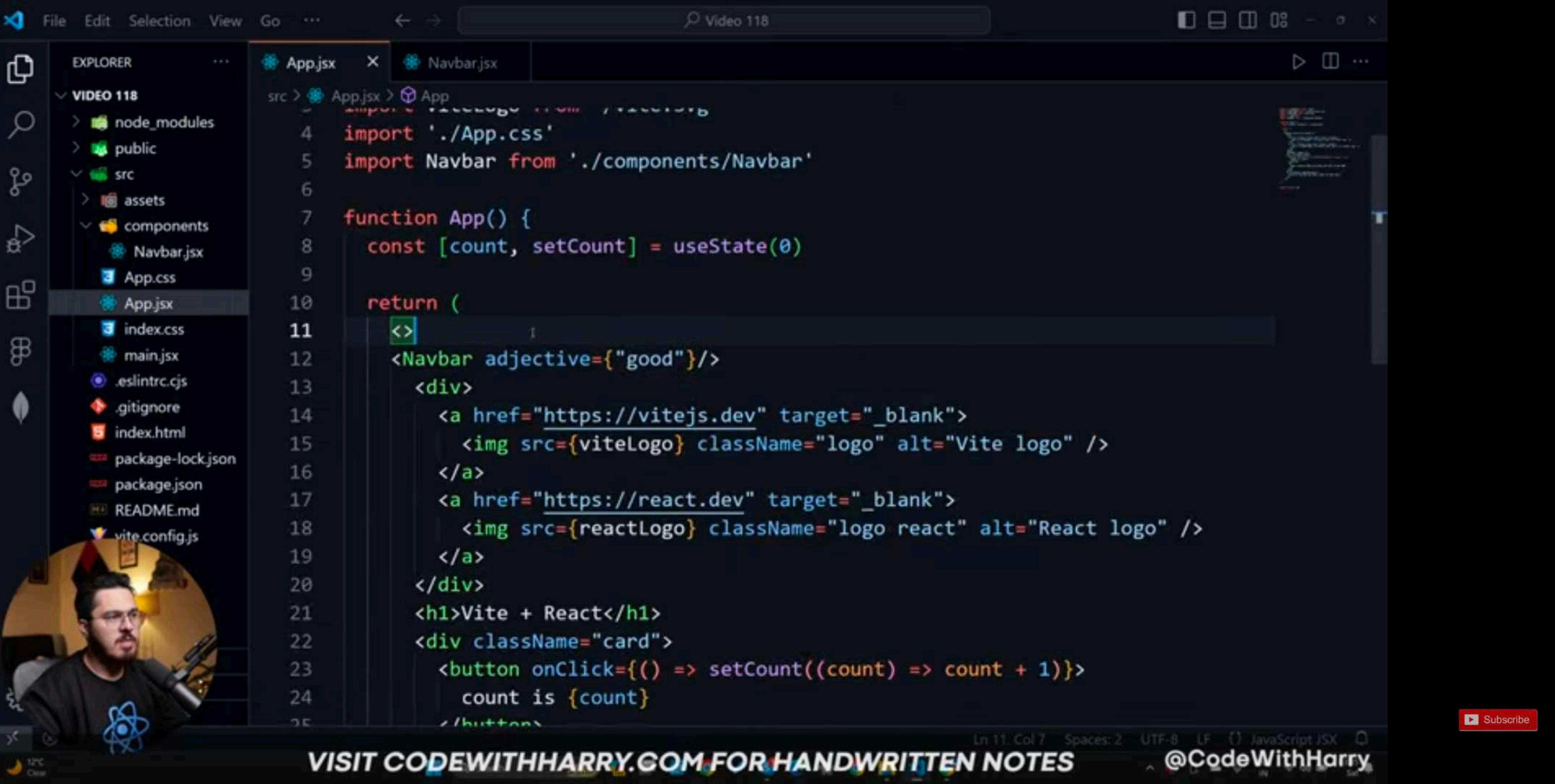


Next: Handling Forms + Connecting React to Ex...



Today's Special • 2/4





File Edit Selection View Go ... ← → Video 118 □ □ □ □ - □ ×

EXPLORER ...

VIDEO 118

- > node_modules
- > public
- > src
 - > assets
 - > components
 - Navbar.jsx
 - App.css
 - App.jsx
 - index.css
 - main.jsx
 - .eslintrc.cjs
 - .gitignore
 - index.html
 - package-lock.json
 - package.json
 - README.md
 - vite.config.js



App.jsx X Navbar.jsx

```
src > App.jsx > App
4 import './App.css'
5 import Navbar from './components/Navbar'

7 function App() {
8   const [count, setCount] = useState(0)
9   const [adjective, setAdjective] = useState("good")

11  const getAdjective = ()=>{
12    return "another"
13  }

15  return (
16    <>
17    <Navbar adjective={"good"} getAdjective={getAdjective} />
18    <div>
19      <a href="https://vitejs.dev" target="_blank">
20        <img src={viteLogo} className="logo" alt="Vite logo" />
21      </a>
22      <a href="https://react.dev" target="_blank">
23        <img src={reactLogo} className="logo react" alt="React logo" />
24      </a>
25    </div>

```

Ln 17, Col 58 Spaces: 2 UTF-8 LF {} JavaScript JSX

12°C Clear Search

Subscribe

03:05 Sat

File Edit Selection View Go ... ← → Video 118 □ □ □ 00 - o x

EXPLORER ...

VIDEO 118

- > node_modules
- > public
- src
- > assets
- components
- Navbar.jsx
- App.css
- App.jsx
- index.css
- main.jsx
- .eslintrc.cjs
- .gitignore
- index.html
- package-lock.json
- package.json
- README.md
- vite.config.js

src > components > Navbar.jsx > [e] default

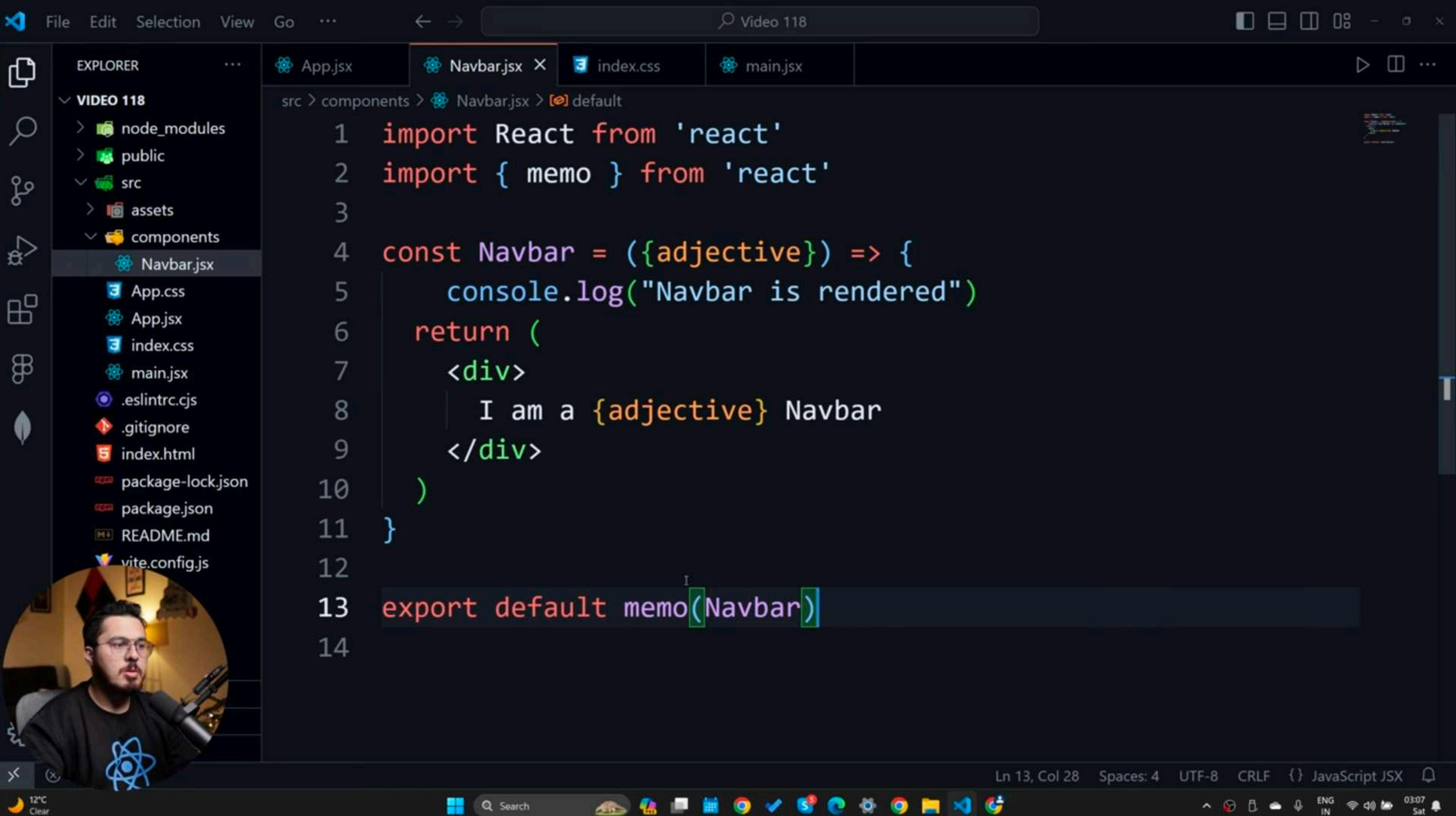
```
1 import React from 'react'
2 import { memo } from 'react'
3
4 const Navbar = ({adjective}) => {
5   console.log("Navbar is rendered")
6   return (
7     <div>
8       I am a {adjective} Navbar
9     </div>
10  )
11 }
12
13 export default memo(Navbar)
14
```

12°C Clear

Search

Ln 13, Col 28 Spaces: 4 UTF-8 CRLF { } JavaScript JSX

Subscribe



File Edit Selection View Go ... ← → Video 118 □ □ □ 00 - □ ×

EXPLORER ...

VIDEO 118

- > node_modules
- > public
- src
- > assets
- components
- Navbar.jsx
- App.css
- App.jsx
- index.css
- main.jsx
- .eslintrc.cjs
- .gitignore
- index.html
- package-lock.json
- package.json
- README.md
- vite.config.js

src > components > Navbar.jsx > [e] Navbar

```
1 import React from 'react'
2 import { memo } from 'react'
3
4 const Navbar = ({adjective, getAdjective}) => {
5   console.log("Navbar is rendered")
6   return (
7     <div>
8       I am a {adjective} Navbar
9       <button onClick={()=>{getAdjective()}}></button>
10    </div>
11  )
12}
13
14 export default memo(Navbar)
15
```

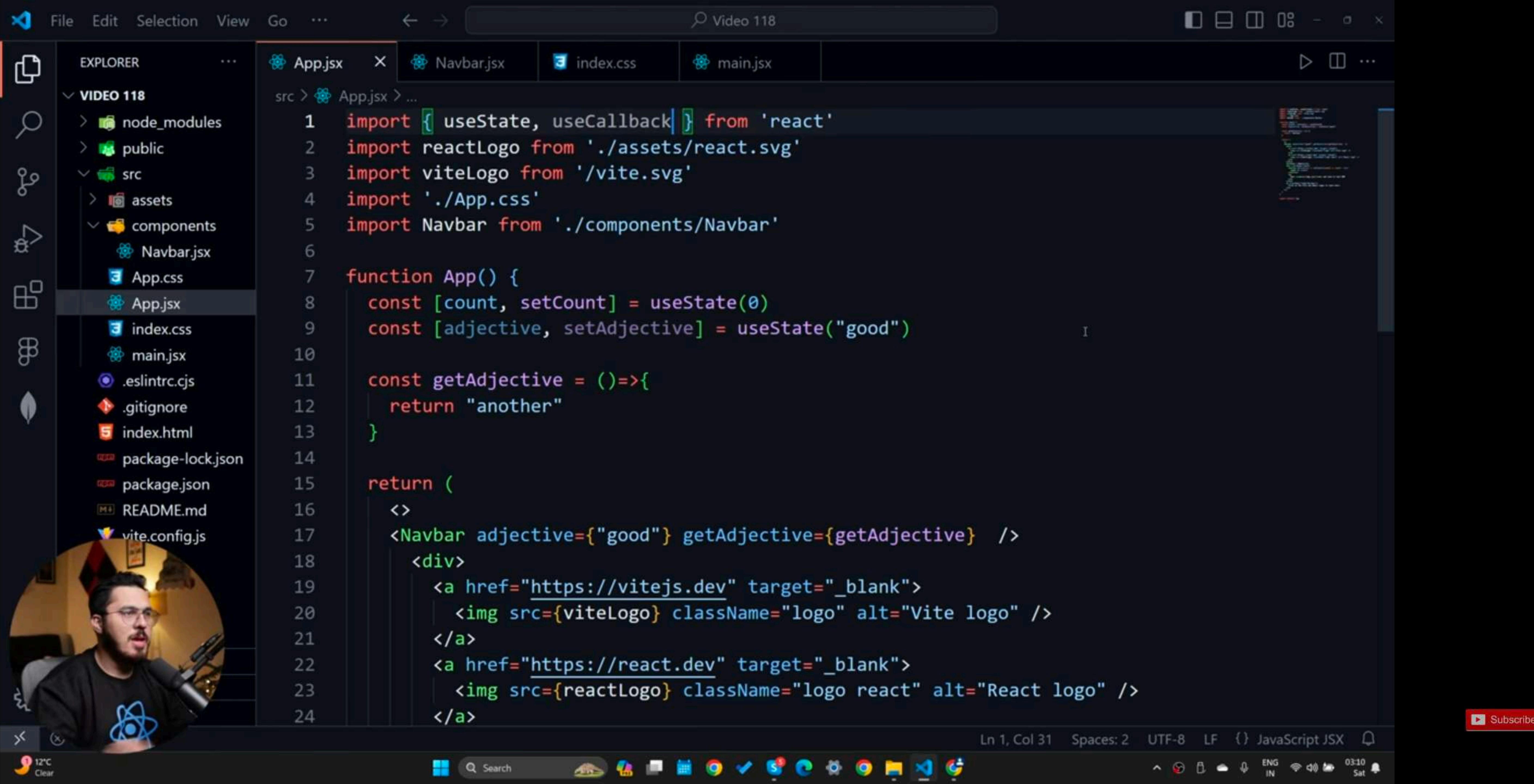
12°C Clear

Search

Ln 9, Col 43 Spaces: 4 UTF-8 CRLF { } JavaScript JSX

Subscribe

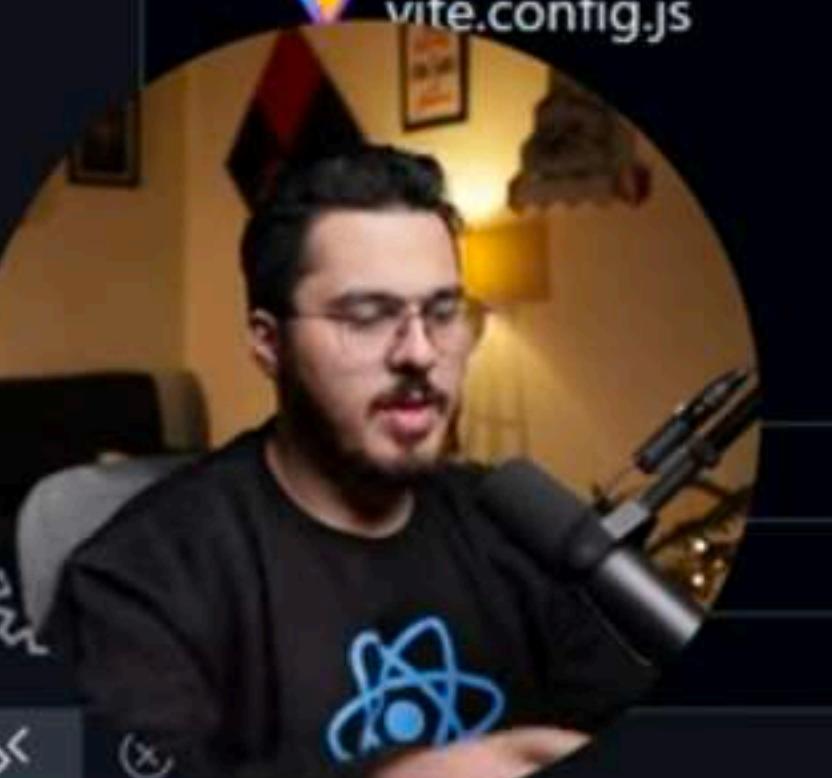




File Edit Selection View Go ... ← → Video 118

EXPLORER

VIDEO 118

- > node_modules
- > public
- > src
 - > assets
 - > components
 - Navbar.jsx
 - App.css
 - App.jsx
 - index.css
 - main.jsx
 - .eslintrc.cjs
 - .gitignore
 - index.html
 - package-lock.json
 - package.json
 - README.md
 - vite.config.js

App.jsx

src > App.jsx > App > getAdjective

```
5 import Navbar from './components/Navbar'
6
7 function App() {
8   const [count, setCount] = useState(0)
9   const [adjective, setAdjective] = useState("good")
10
11  const getAdjective = useCallback(() => {
12    return "another"
13  },
14  [])
15
16
17  return (
18    <>
19      <Navbar adjective={"good"} getAdjective={getAdjective} />
20      <div>
21        <a href="https://vitejs.dev" target="_blank">
22          <img src={viteLogo} className="logo" alt="Vite logo" />
23        </a>
24        <a href="https://react.dev" target="_blank">
25          <img src={reactLogo} className="logo react" alt="React logo" />
26        </a>
27      </div>
28      <h1>Vite + React</h1>

```

Ln 14, Col 7 Spaces: 2 UTF-8 LF {} JavaScript JSX

12°C Clear

Search

03:11 Sat

Subscribe

File Edit Selection View Go ... ← → Video 118

EXPLORER

VIDEO 118

- > node_modules
- > public
- > src
 - > assets
 - > components
 - Navbar.jsx
 - App.css
 - App.jsx
 - index.css
 - main.jsx
- .eslintrc.cjs
- .gitignore
- index.html
- package-lock.json
- package.json
- README.md
- vite.config.js



App.jsx

src > App.jsx > App > getAdjective

```
5 import Navbar from './components/Navbar'
6
7 function App() {
8   const [count, setCount] = useState(0)
9   const [adjective, setAdjective] = useState("good")
10
11  const getAdjective = useCallback(() => {
12    return "another"
13  }, [])
14
15
16
17  return (
18    <>
19      <Navbar adjective={"good"} getAdjective={getAdjective} />
20      <div>
21        <a href="https://vitejs.dev" target="_blank">
22          <img src={viteLogo} className="logo" alt="Vite logo" />
23        </a>
24        <a href="https://react.dev" target="_blank">
25          <img src={reactLogo} className="logo react" alt="React logo" />
26        </a>
27      </div>
28      <h1>Vite + React</h1>

```

Ln 13, Col 5 Spaces: 2 UTF-8 LF {} JavaScript JSX

12°C Clear

Search

Subscribe

File Edit Selection View Go ... ← → Video 118 □ □ □ 00 - o x

EXPLORER ...

VIDEO 118

> node_modules

> public

> src

> assets

> components

Navbar.jsx

App.css

App.jsx

index.css

main.jsx

.eslintrc.cjs

.gitignore

index.html

package-lock.json

package.json

README.md

vite.config.js

12345678910111213141516171819202122232425262728

```
import Navbar from './components/Navbar'
function App() {
  const [count, setCount] = useState(0)
  const [adjective, setAdjective] = useState("good")
  const getAdjective = useCallback(() => {
    return "another"
  }, [])
  return (
    <>
    <Navbar adjective={"good"} getAdjective={getAdjective} />
    <div>
      <a href="https://vitejs.dev" target="_blank">
        <img src={viteLogo} className="logo" alt="Vite logo" />
      </a>
      <a href="https://react.dev" target="_blank">
        <img src={reactLogo} className="logo react" alt="React logo" />
      </a>
    </div>
    <h1>Vite + React</h1>
    <div className="card">
```

Ln 21, Col 28 (350 selected) Spaces: 2 UTF-8 LF { JavaScript JSX

12°C Clear

Search

Subscribe

03:11 Sat

ENG IN

4:53



1.74 MB/s 5G 89%



Visit advertiser

Sponsored ⓘ



Power Up Your Career



Sponsored · sawan.sgacadem...

Sign up

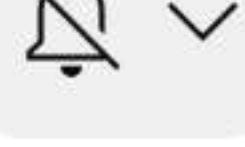
Handling Forms + Connecting React to Express Backend | Sigma Web Development..

91K views 1y ago Sigma Web Development Course - Web ...more



CodeWithHarry 7.98M

Join



2.7K

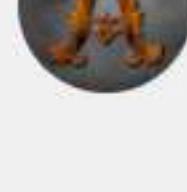


Share

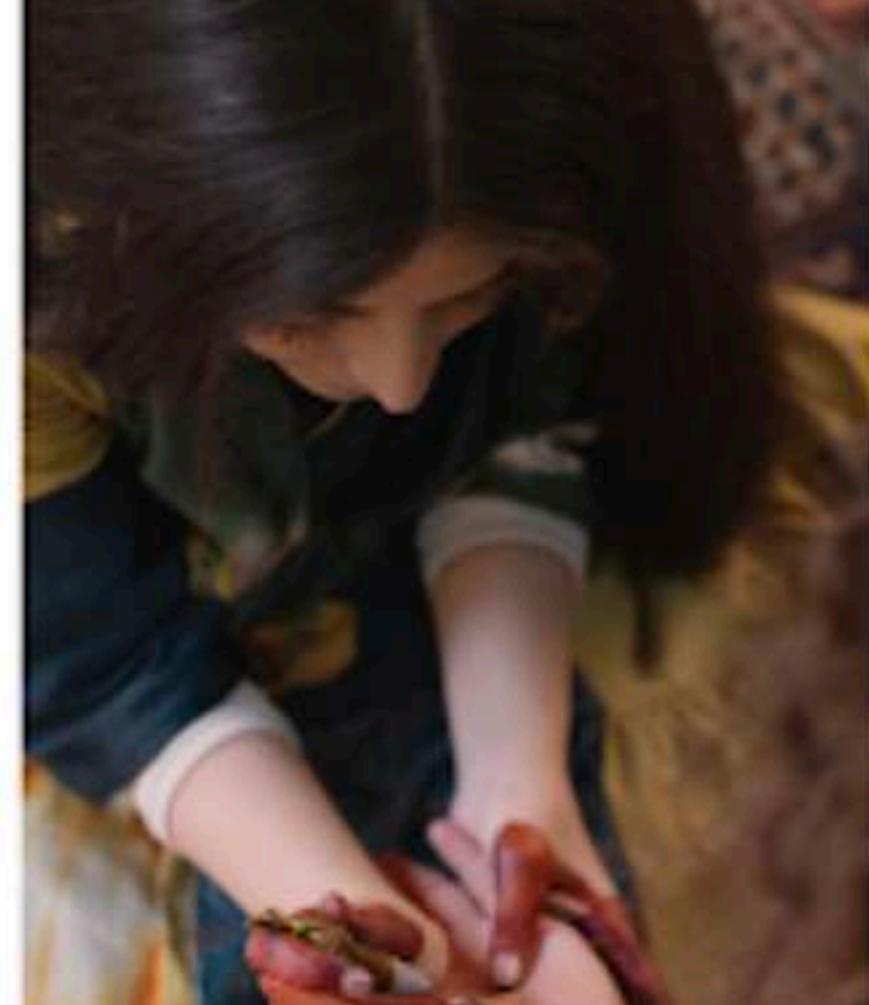
Remix

Stop ads

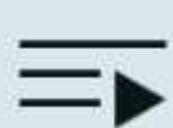
Comments 243



Grow a little bit more beard sir, you will look like Dennis Ritchie. 😊



Next: Learn Redux in One video | Sigma Web De...



Today's Special · 2/3



Sigma Web Development Course - VI React Hook Form - performant, flexi...

react-hook-form.com

Sigma Web Developme... GitHub - CodeWithHarr... Learn to code online - C... Es with Express

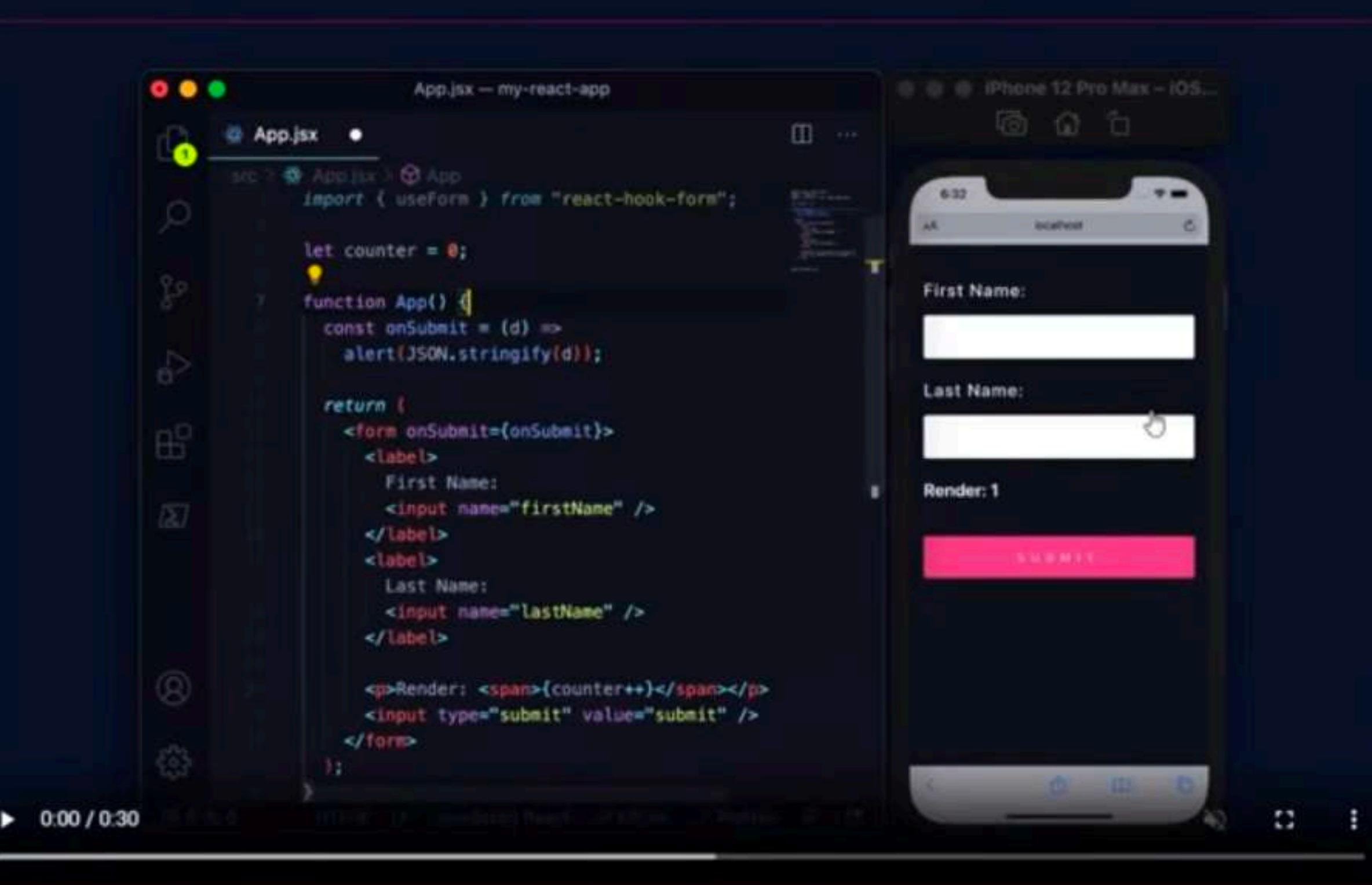
Search... 

Home Get Started API TS Advanced FAQs Tools ▾ Resources ▾ Releases

 **React Hook Form**

Performant, flexible and extensible forms with easy-to-use validation.

[Demo](#) [Get Started ▶](#)



App.jsx — my-react-app

```
App.jsx
import { useForm } from "react-hook-form";

let counter = 0;

function App() {
  const onSubmit = (d) =>
    alert(JSON.stringify(d));

  return (
    <form onSubmit={onSubmit}>
      <label>
        First Name:
        <input name="firstName" />
      </label>
      <label>
        Last Name:
        <input name="lastName" />
      </label>

      <p>Render: <span>{counter++}</span></p>
      <input type="submit" value="submit" />
    </form>
  );
}

export default App;
```

iPhone 12 Pro Max - iOS...

6:32 AM localhost

First Name:

Last Name:

Render: 1

Submit

0:00 / 0:30



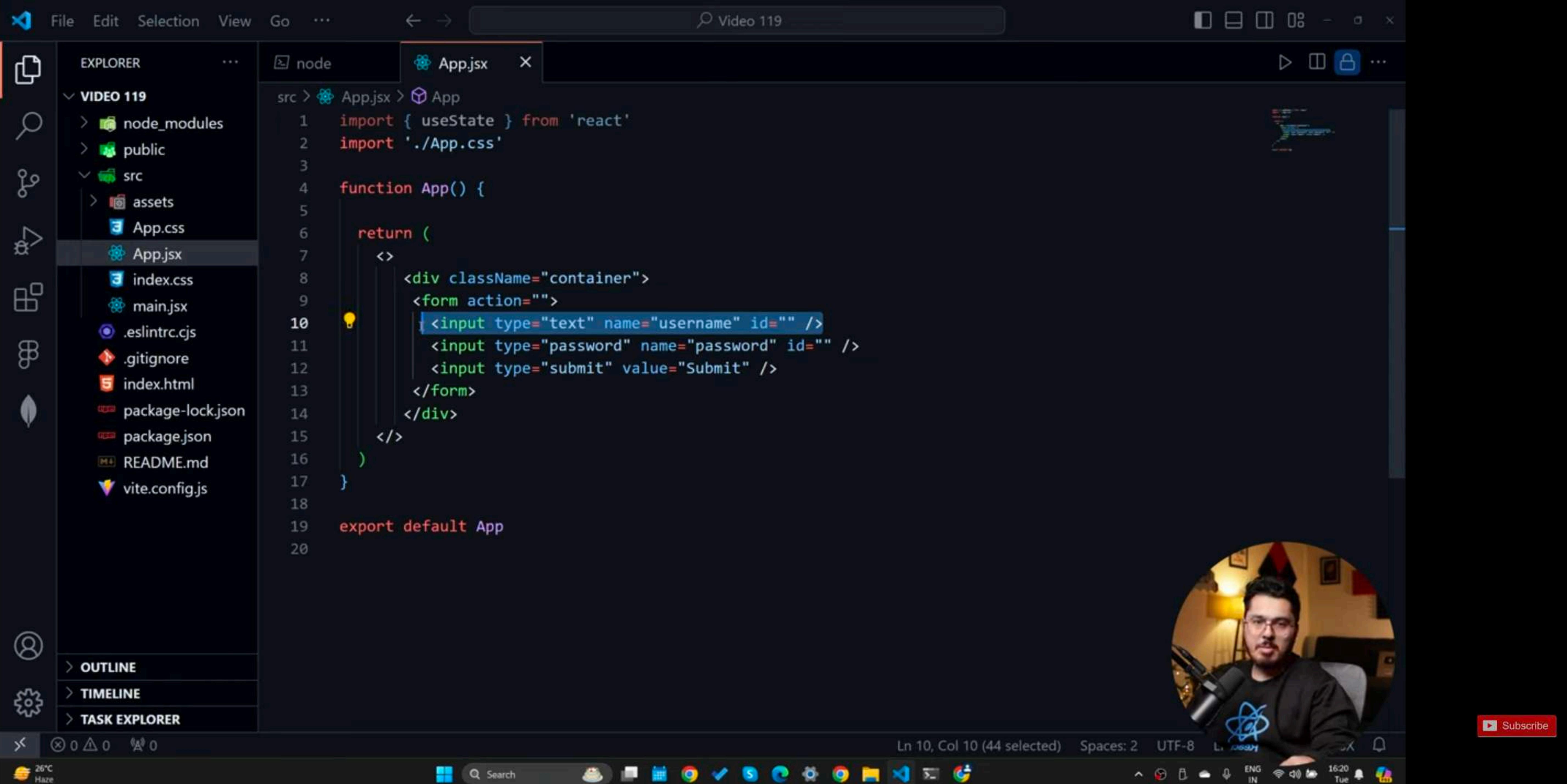
Subscribe

26°C Haze

Search

16:18 Tue

ENG IN



Get Started

Simple form validation with React Hook Form

Installation

Installing React Hook Form only takes a single command and you're ready to roll.

```
npm install react-hook-form
```

COPIED

Example

The following code excerpt demonstrates a basic usage example:

5

```
import { useForm } from "react-hook-form"

export default function App() {
  const {
    register,
    handleSubmit,
    watch,
    formState: { errors },
  } = useForm()

  const onSubmit = (data) => console.log(data)

  console.log(watch("example")) // watch input value by passing the name of it

  return (
    /* "handleSubmit" will validate your inputs before invoking "onSubmit" */
    <form onSubmit={handleSubmit(onSubmit)}>
      /* register your input into the hook by invoking the "register" function
       * and passing it to the "input" component
      <input defaultValue="test" {...register("example")}>

      /* include validation with required or other standard HTML validation rules
       * and pass it to the "input" component
      <input {...register("exampleRequired", { required: true })}>
      /* errors will return when field validation fails */
      {errors.exampleRequired?.message} This field is required
    </form>
  )
}
```

COPY CODESANDBOX[®]



Sigma Web Development Course - W Get Started react-hook-form.com/get-started Sigma Web Developme... GitHub - CodeWithHarr... Learn to code online - C... Es with Express

Installing React Hook Form only takes a single command and you're ready to roll.

```
npm install react-hook-form
```

COPY

Example

The following code excerpt demonstrates a basic usage example:

Menu

- Quick start
- React Web Video Tutorial
- Register fields
- Apply validation
- Integrating an existing form
- Integrating with UI libraries
- Integrating Controlled Inputs
- Integrating with global state
- Integrating with services
- Handle errors
- Schema Validation
- React Native
- TypeScript
- Design and philosophy

TS

```
import { useForm } from "react-hook-form"

export default function App() {
  const {
    register,
    handleSubmit,
    watch,
    formState: { errors },
  } = useForm()

  const onSubmit = (data) => console.log(data)

  console.log(watch("example")) // watch input value by passing the name of it

  return (
    /* "handleSubmit" will validate your inputs before invoking "onSubmit" */
    <form onSubmit={handleSubmit(onSubmit)}>
      /* register your input into the hook by invoking the "register" function */
      <input defaultValue="test" {...register("example")} />

      /* include validation with required or other standard HTML validation rules */
      <input {...register("exampleRequired", { required: true })} />
      /* errors will return when field validation fails */
      {errors.exampleRequired && <span>This field is required</span>}

      <input type="submit" />
    </form>
  )
}
```

JS

COPY CODESANDBOX

React Web Video Tutorial

This video tutorial illustrates the basic usage and concepts of React Hook Form.

React Hook Form - Get Started

Share

26°C Haze

Search

16:21 Tue

ENG IN

Subcribe



File Edit Selection View Go ... ← → Video 119 □ □ □ □ - □ × □ □ □ □ ...

EXPLORER ...

VIDEO 119

> node_modules

> public

> src

> assets

App.css

App.jsx

index.css

main.jsx

.eslintrc.cjs

.gitignore

index.html

package-lock.json

package.json

README.md

vite.config.js

OUTLINE

TIMELINE

TASK EXPLORER

node App.jsx

```
src > App.jsx > App
1 import { useState } from 'react'
2 import './App.css'
3 import { useForm } from "react-hook-form"
4
5
6
7 function App() {
8   const [
9     register,
10    handleSubmit,
11    watch,
12    formState: { errors },
13  } = useForm()
14
15  return (
16    <>
17      <div className="container">
18        <form action="">
19          <input type="text" name="username" id="" />
20          <input type="password" name="password" id="" />
21          <input type="submit" value="Submit" />
22        </form>
23      </div>
24    </>
25  )
26
27
28 export default App
```

Ln 9, Col 9 Spaces: 2 UTF-8 L. 0699

26°C Haze

Search

16:22 Tue

ENG IN

Subscribe



Sigma Web Development Course - VI | Get Started

react-hook-form.com/get-started

Sigma Web Developme... GitHub - CodeWithHarr... Learn to code online - C... Es with Express

Installing React Hook Form only takes a single command and you're ready to roll.

```
npm install react-hook-form
```

COPY

Example

The following code excerpt demonstrates a basic usage example:

Menu

- Quick start
- React Web Video Tutorial
- Register fields
- Apply validation
- Integrating an existing form
- Integrating with UI libraries
- Integrating Controlled Inputs
- Integrating with global state
- Integrating with services
- Handle errors
- Schema Validation
- React Native
- TypeScript
- Design and philosophy

TS

```
import { useForm } from "react-hook-form"

export default function App() {
  const {
    register,
    handleSubmit,
    watch,
    formState: { errors },
  } = useForm()

  const onSubmit = (data) => console.log(data)

  console.log(watch("example")) // watch input value by passing the name of it

  return (
    /* "handleSubmit" will validate your inputs before invoking "onSubmit" */
    <form onSubmit={handleSubmit(onSubmit)}>
      /* register your input into the hook by invoking the "register" function */
      <input defaultValue="test" {...register("example")} />

      /* include validation with required or other standard HTML validation rules */
      <input {...register("exampleRequired", { required: true })} />
      /* errors will return when field validation fails */
      {errors.exampleRequired && <span>This field is required</span>}

      <input type="submit" />
    </form>
  )
}
```

JS

COPY CODESANDBOX

React Web Video Tutorial

This video tutorial illustrates the basic usage and concepts of React Hook Form.

React Hook Form - Get Started

Share

26°C Haze

Search

16:22 Tue

ENG IN

Subcribe

File Edit Selection View Go ... ← → Video 119 □ □ □ 00 - o x □ □ □ □ □ ...

EXPLORER ... node App.jsx X

VIDEO 119

node_modules

public

src

assets

App.css

App.jsx

index.css

main.jsx

.eslintrc.cjs

.gitignore

index.html

package-lock.json

package.json

README.md

vite.config.js

OUTLINE

TIMELINE

TASK EXPLORER

26°C Haze 0 △ 0 ⌂ 0

src > App.jsx > App

```
1 import { useState } from 'react'
2 import './App.css'
3 import { useForm } from "react-hook-form"
4
5
6
7 function App() {
8   const {
9     register,
10    handleSubmit,
11    watch,
12    formState: { errors },
13  } = useForm();
14
15  return (
16    <>
17      <div className="container">
18        <form action="" onSubmit={handleSubmit(onSubmit)}>
19          <input type="text" name="username" id="" />
20          <input type="password" name="password" id="" />
21          <input type="submit" value="Submit" />
22        </form>
23      </div>
24    </>
25  )
26}
27
28 export default App
```

Ln 18, Col 58 Spaces: 2 UTF-8 L. 0:00:00

26°C Haze 0 △ 0 ⌂ 0

Search

16:22 Tue ENG IN

Subscribe



Sigma Web Development Course - VI | Get Started | react-hook-form.com/get-started | Sigma Web Developme... | GitHub - CodeWithHarr... | Learn to code online - C... | Ets with Express

Installing React Hook Form only takes a single command and you're ready to roll.

```
npm install react-hook-form
```

COPY

Example

The following code excerpt demonstrates a basic usage example:

Menu

- Quick start
- React Web Video Tutorial
- Register fields
- Apply validation
- Integrating an existing form
- Integrating with UI libraries
- Integrating Controlled Inputs
- Integrating with global state
- Integrating with services
- Handle errors
- Schema Validation
- React Native
- TypeScript
- Design and philosophy

TS

```
import { useForm } from "react-hook-form"

export default function App() {
  const {
    register,
    handleSubmit,
    watch,
    formState: { errors },
  } = useForm()

  const onSubmit = (data) => console.log(data)

  console.log(watch("example")) // watch input value by passing the name of it

  return (
    /* "handleSubmit" will validate your inputs before invoking "onSubmit" */
    <form onSubmit={handleSubmit(onSubmit)}>
      /* register your input into the hook by invoking the "register" function */
      <input defaultValue="test" {...register("example")} />

      /* include validation with required or other standard HTML validation rules */
      <input {...register("exampleRequired", { required: true })} />
      /* errors will return when field validation fails */
      {errors.exampleRequired && <span>This field is required</span>}

      <input type="submit" />
    </form>
  )
}
```

JS

COPY CODESANDBOX

React Web Video Tutorial

This video tutorial illustrates the basic usage and concepts of React Hook Form.

React Hook Form - Get Started

Share

26°C Haze

Search

16:22 Tue

ENG IN

Subscribe



File Edit Selection View Go ... ← → Video 119 □ □ □ 00 - o x

EXPLORER ... node App.jsx X

VIDEO 119

- > node_modules
- > public
- src
 - > assets
 - App.css
 - App.jsx
 - index.css
 - main.jsx
 - .eslintrc.cjs
 - .gitignore
 - index.html
 - package-lock.json
 - package.json
 - README.md
 - vite.config.js

OUTLINE

Timeline

Task Explorer

```
src > App.jsx > App > [o] onSubmit
1 import { useState } from 'react'
2 import './App.css'
3 import { useForm } from "react-hook-form"
4
5
6
7 function App() {
8   const {
9     register,
10    handleSubmit,
11    watch,
12    formState: { errors },
13  } = useForm();
14
15  const onSubmit = (data) => console.log(data)
16
17  return (
18    <>
19      <div className="container">
20        <form action="" onSubmit={handleSubmit(onSubmit)}>
21          <input type="text" name="username" id="" />
22          <input type="password" name="password" id="" />
23          <input type="submit" value="Submit" />
24        </form>
25      </div>
26    </>
27  )
28 }
```

Ln 15, Col 47 Spaces: 2 UTF-8 L. 16:22 Tue ENG IN

26°C Haze

Search

16:22 Tue ENG IN

Subscribe



Sigma Web Development Course - VI | Get Started | react-hook-form.com/get-started | Sigma Web Development | GitHub - CodeWithHarr... | Learn to code online - C... | Ejs with Express

Installing React Hook Form only takes a single command and you're ready to roll.

```
npm install react-hook-form
```

COPY

Example

The following code excerpt demonstrates a basic usage example:

Menu

- Quick start
- React Web Video Tutorial
- Register fields
- Apply validation
- Integrating an existing form
- Integrating with UI libraries
- Integrating Controlled Inputs
- Integrating with global state
- Integrating with services
- Handle errors
- Schema Validation
- React Native
- TypeScript
- Design and philosophy

TS

```
import { useForm } from "react-hook-form"

export default function App() {
  const {
    register,
    handleSubmit,
    watch,
    formState: { errors },
  } = useForm()

  const onSubmit = (data) => console.log(data)

  console.log(watch("example")) // watch input value by passing the name of it

  return (
    /* "handleSubmit" will validate your inputs before invoking "onSubmit" */
    <form onSubmit={handleSubmit(onSubmit)}>
      /* register your input into the hook by invoking the "register" function */
      <input defaultValue="test" {...register("example")}>

      /* include validation with required or other standard HTML validation rules */
      <input {...register("exampleRequired", { required: true })}>
      /* errors will return when field validation fails */
      {errors.exampleRequired && <span>This field is required</span>}

      <input type="submit" />
    </form>
  )
}
```

JS

COPY CODESANDBOX

React Web Video Tutorial

This video tutorial illustrates the basic usage and concepts of React Hook Form.

React Hook Form - Get Started

Share

26°C Haze

Search

16:23 Tue

ENG IN

Subscribe



File Edit Selection View Go ... ← → Video 119 □ □ □ 00 - o x □ □ □ ...

EXPLORER ... node App.jsx X

VIDEO 119

src > App.jsx > App

```
6
7  function App() {
8    const {
9      register,
10     handleSubmit,
11     watch,
12     formState: { errors },
13   } = useForm();
14
15  const onSubmit = (data) => console.log(data)
16
17  return (
18    <>
19      <div className="container">
20        <form action="" onSubmit={handleSubmit(onSubmit)}>
21          <input {...register("username")} type="text" name="username" id="" />
22          <input {...register("password")} type="password" name="password" id="" />
23          <input type="submit" value="Submit" />
24        </form>
25      </div>
26    </>
27  )
28
29
30  export default App
31
```

OUTLINE

TIMELINE

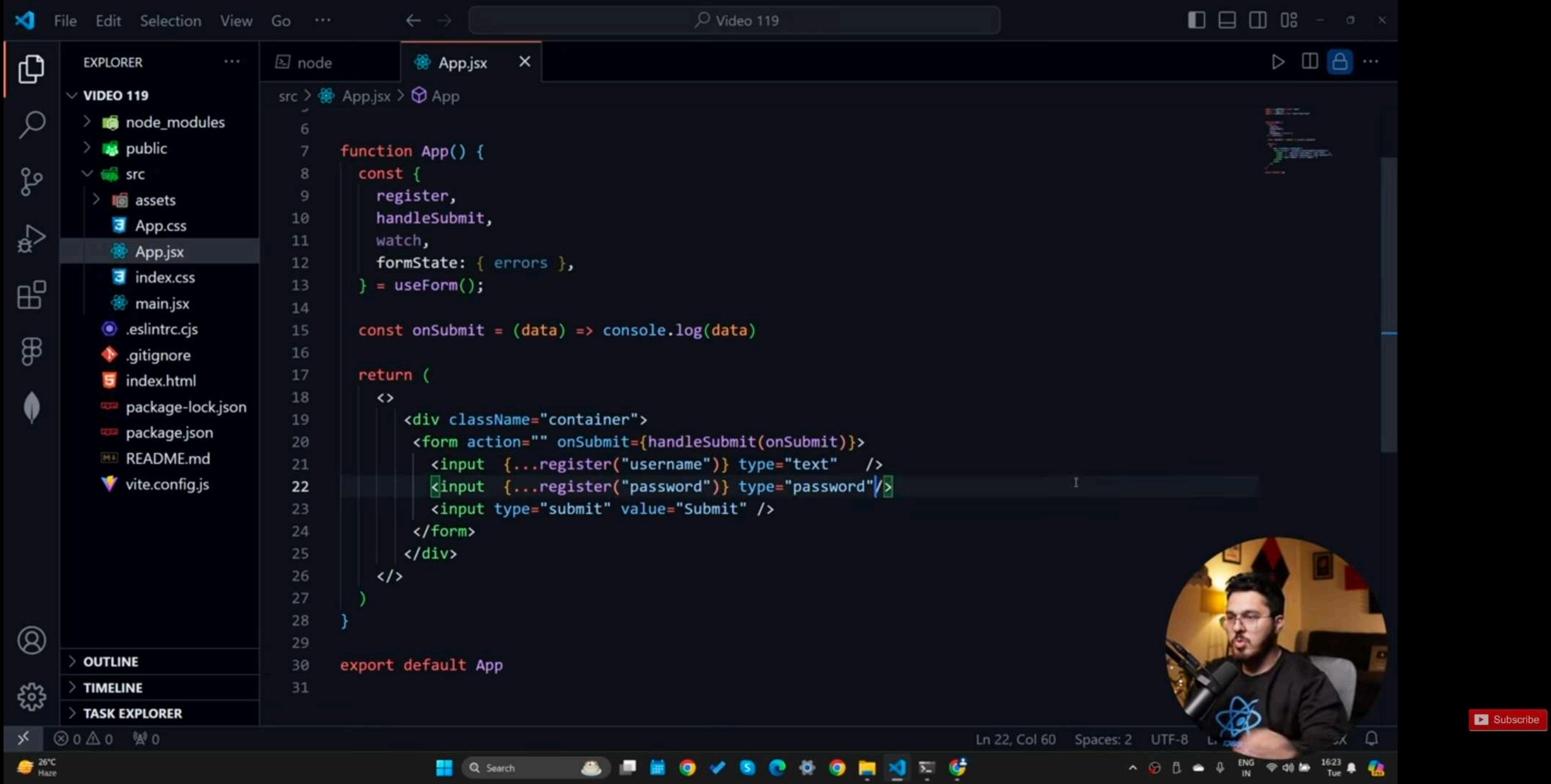
TASK EXPLORER

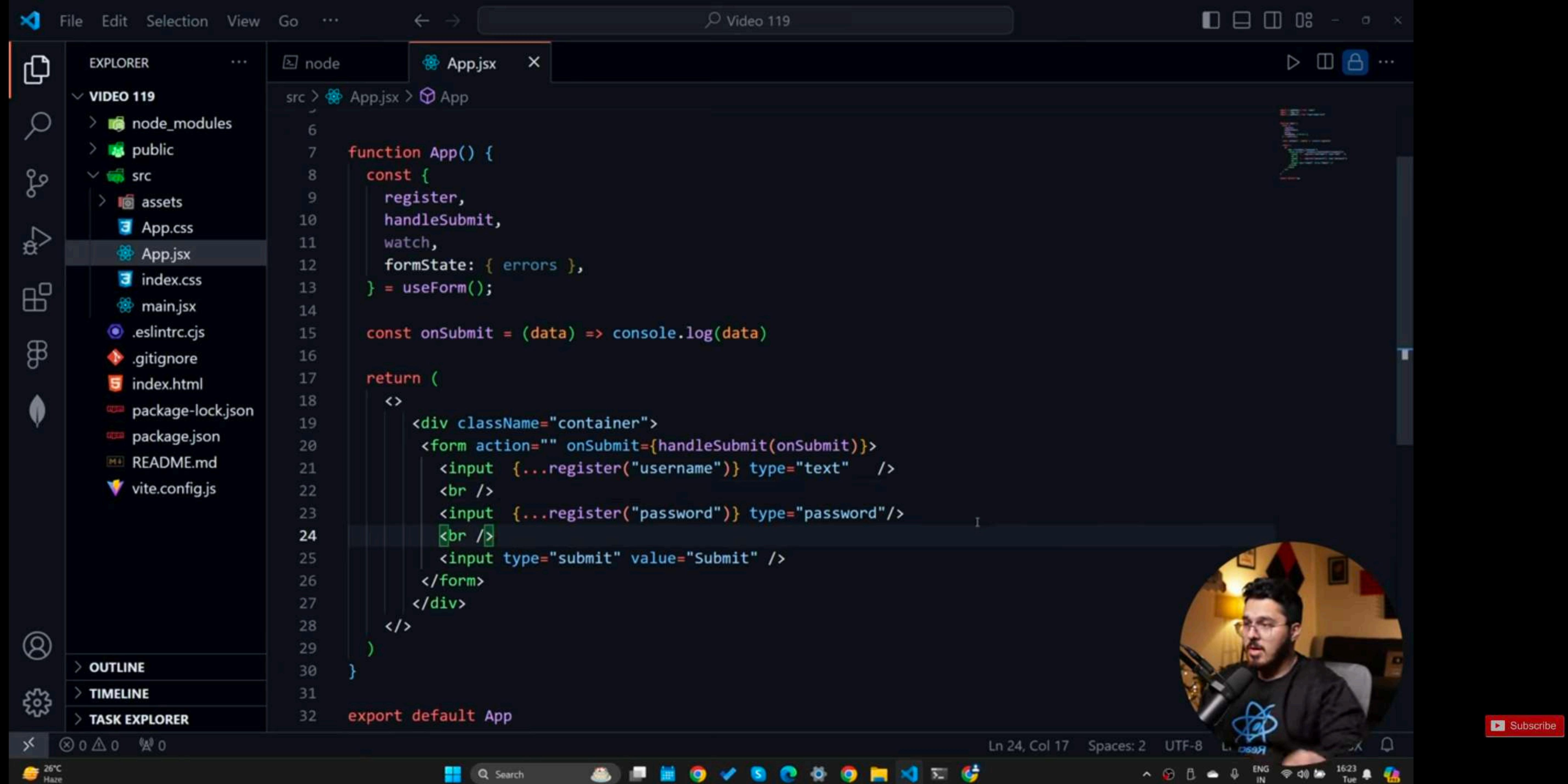
26°C Haze 0 △ 0 ⌂ 0

Search

Ln 22, Col 41 Spaces: 2 UTF-8 L. 16:23 Tue ENG IN

Subscribe





File Edit Selection View Go ... ← → Video 119 □ □ □ 00 - o x □ □ □ 000 ...

EXPLORER ...

VIDEO 119

- > node_modules
- > public
- src
 - > assets
 - App.css
 - App.jsx
 - index.css
 - main.jsx
 - .eslintrc.cjs
 - .gitignore
 - index.html
 - package-lock.json
 - package.json
 - README.md
 - vite.config.js

OUTLINE

TIMELINE

TASK EXPLORER

node App.jsx

```
src > App.jsx > App > maxLength
7  function App() {
8    const {
9      register,
10     handleSubmit,
11     watch,
12     formState: { errors },
13   } = useForm();
14
15  const onSubmit = (data) => console.log(data)
16
17  return (
18    <>
19      <div className="container">
20        <form action="" onSubmit={handleSubmit(onSubmit)}>
21          <input placeholder='username' {...register("username", { required: true, minLength: {value: 3, message: "Min length is 3"}, maxLength: {value: 8, message: "Max length is 8"} })} type="text" />
22          {errors.username && <div className='red'>{errors.username.message}</div>}
23          <br />
24          <input placeholder='password' {...register("password")}> type="password"/>
25          <br />
26          <input type="submit" value="Submit" />
27        </form>
28      </div>
29    </>
30  )
31
32 }
```

Ln 21, Col 146 Spaces: 2 UTF-8 L. 26°C Mostly sunny

Search

26°C Mostly sunny

16:40 Tue ENG IN

Subscribe



File Edit Selection View Go ... ← → Video 119 □ □ □ 00 - o x □ □ □ 00 ...

EXPLORER ... node App.jsx X

VIDEO 119

node_modules

public

src

assets

App.css

App.jsx

index.css

main.jsx

.eslintrc.cjs

.gitignore

index.html

package-lock.json

package.json

README.md

vite.config.js

OUTLINE

TIMELINE

TASK EXPLORER

src > App.jsx > App > onSubmit

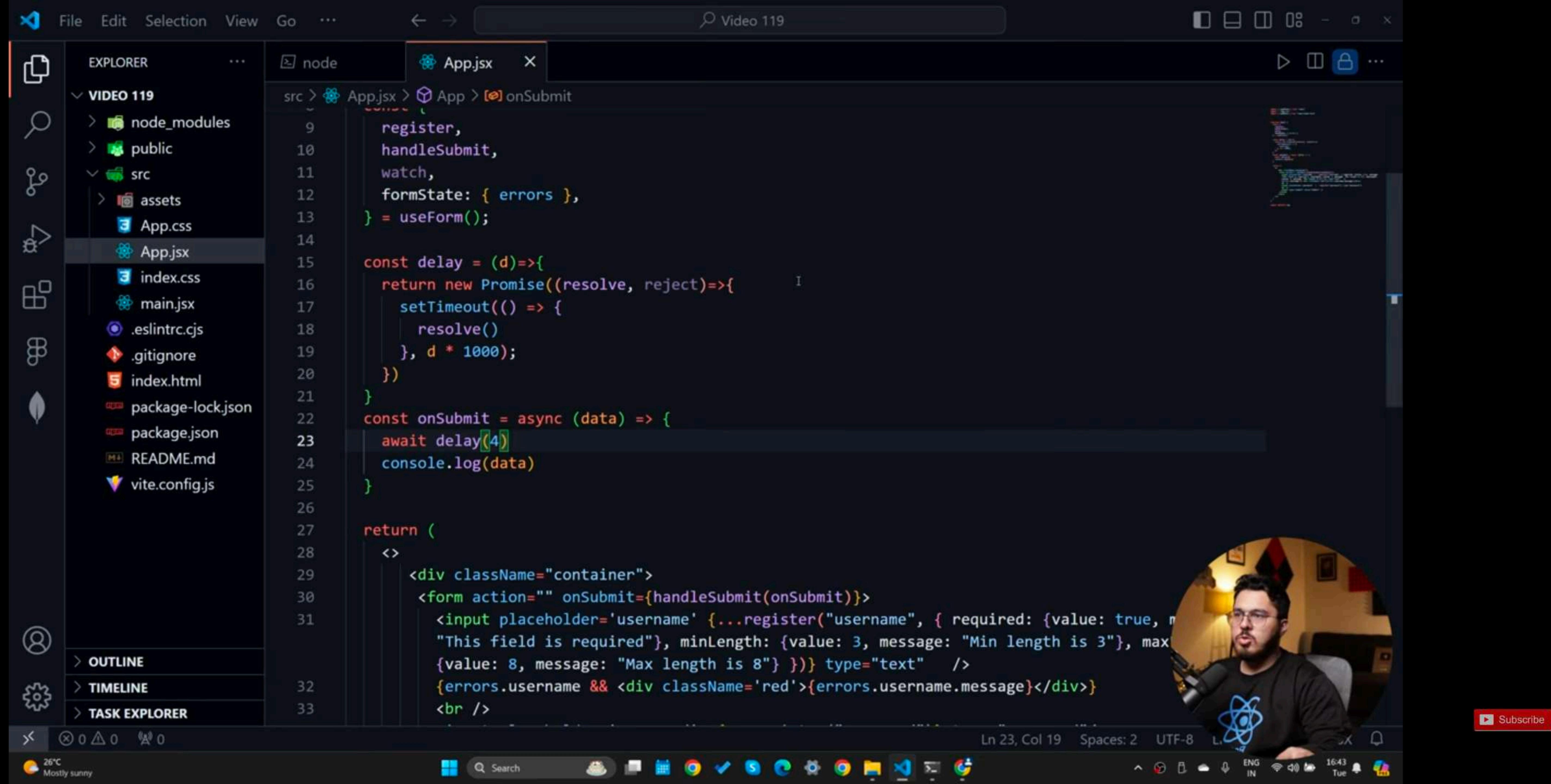
```
6
7 function App() {
8   const {
9     register,
10    handleSubmit,
11    watch,
12    formState: { errors },
13  } = useForm();
14
15  const onSubmit = (data) => {console.log(data)}
16
17  return (
18    <>
19      <div className="container">
20        <form action="" onSubmit={handleSubmit(onSubmit)}>
21          <input placeholder='username' {...register("username", { required: {value: true, message: "This field is required"}, minLength: {value: 3, message: "Min length is 3"}, maxLength: {value: 8, message: "Max length is 8"} })} type="text" />
22          {errors.username && <div className='red'>{errors.username.message}</div>}
23          <br />
24          <input placeholder='password' {...register("password")}> type="password"/>
25          <br />
26          <input type="submit" value="Submit" />
27        </form>
28      </div>
29    </>
30  )
}
```

Ln 15, Col 48 (17 selected) Spaces: 2 UTF-8 L 26°C Mostly sunny 16:42 Tue ENG IN

Search



Subscribe



File Edit Selection View Go ... ← → Video 119 □ □ □ 00 - o x □ □ □ □ □ ...

EXPLORER ... node App.jsx X

VIDEO 119

node_modules

public

src

assets

App.css

App.jsx

index.css

main.jsx

.eslintrc.cjs

.gitignore

index.html

package-lock.json

package.json

README.md

vite.config.js

OUTLINE

TIMELINE

TASK EXPLORER

src > App.jsx > App

```
21 }
22
23 const onSubmit = async (data) => {
24   await delay(2) // simulating network delay
25   console.log(data)
26 }
27
28 return (
29   <>
30   <div className="container">
31     <form action="" onSubmit={handleSubmit(onSubmit)}>
32       <input placeholder='username' {...register("username", { required: {value: true, message: "This field is required"}, minLength: {value: 3, message: "Min length is 3"}, maxLength: {value: 8, message: "Max length is 8"} })} type="text" />
33       {errors.username && <div className='red'>{errors.username.message}</div>}
34       <br />
35       <input placeholder='password' {...register("password")}> type="password"/>
36       <br />
37       <input disabled={isSubmitting} type="submit" value="Submit" />
38     </form>
39   </div>
40 )
41
42 }
43
44 export default App
```

Ln 37, Col 40 Spaces: 2 UTF-8 L 26°C Mostly sunny 16:44 Tue ENG IN

Search



Subscribe

File Edit Selection View Go ... ← → Video 119 □ □ □ 00 - o x

EXPLORER ... node App.jsx X

VIDEO 119

- > node_modules
- > public
- src
 - > assets
 - App.css
 - App.jsx
 - index.css
 - main.jsx
 - .eslintrc.cjs
 - .gitignore
 - index.html
 - package-lock.json
 - package.json
 - README.md
 - vite.config.js

OUTLINE

TIMELINE

TASK EXPLORER

26°C Mostly sunny

Search

Ln 11, Col 11 Spaces: 2 UTF-8

20 16:45 Tue ENG IN

21 16:45 Tue ENG IN

22 16:45 Tue ENG IN

23 16:45 Tue ENG IN

24 16:45 Tue ENG IN

25 16:45 Tue ENG IN

26 16:45 Tue ENG IN

27 16:45 Tue ENG IN

28 16:45 Tue ENG IN

```
src > App.jsx > App
1 import { useState } from 'react'
2 import './App.css'
3 import { useForm } from "react-hook-form"
4
5
6
7 function App() {
8   const [
9     register,
10    handleSubmit,
11    watch,
12
13    formState: { errors, isSubmitting },
14  ] = useForm();
15
16  const delay = (d)=>{
17    return new Promise((resolve, reject)=>{
18      setTimeout(() => {
19        resolve()
20      }, d * 1000);
21    })
22
23
24  const onSubmit = async (data) => {
25    await delay(2) // simulating network delay
26    console.log(data)
27
28 }
```



File Edit Selection View Go ... ← → Video 119 □ □ □ 00 - o x

EXPLORER ... node App.jsx X

VIDEO 119

src > App.jsx > App

```
22
23     const onSubmit = async (data) => {
24         await delay(2) // simulating network delay
25         console.log(data)
26     }
27
28     return (
29         <>
30         {isSubmitting && <div>Loading...</div>}
31         <div className="container"> ⚡ Loading...
32             <form action="" onSubmit={handleSubmit(onSubmit)}>
33                 <input placeholder='username' {...register("username", { required: { value: true, message: "This field is required" }, minLength: { value: 3, message: "Min length is 3" }, maxLength: { value: 8, message: "Max length is 8" } })} type="text" />
34                 {errors.username && <div className='red'>{errors.username.message}</div>}
35                 <br />
36                 <input placeholder='password' {...register("password")}> type="password"/>
37                 <br />
38                 <input disabled={isSubmitting} type="submit" value="Submit" />
39             </form>
40         </div>
41     )
42 )
43
44
45     export default App
46
```

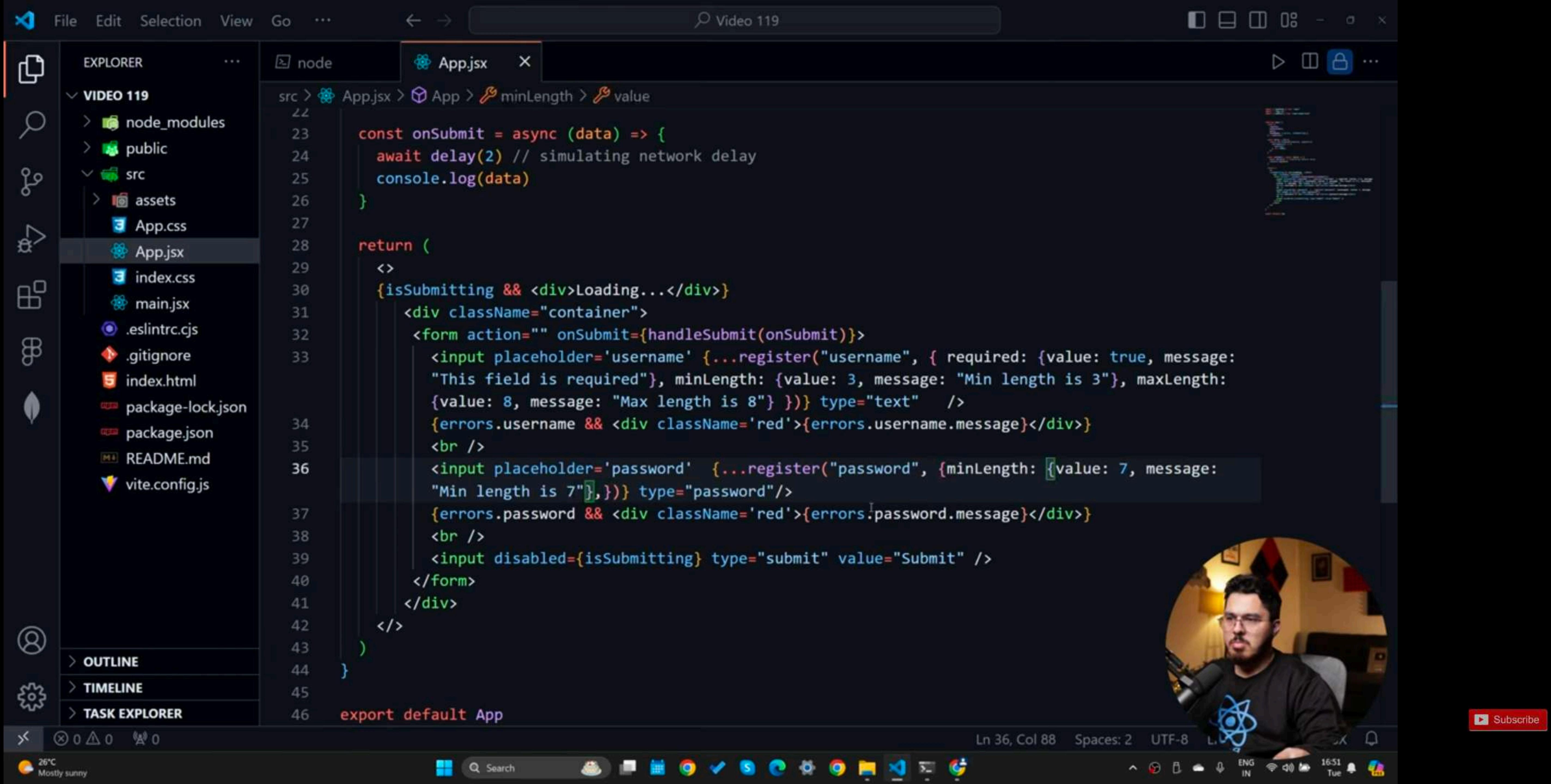
Emmet Abbreviation ⚡ Loading...

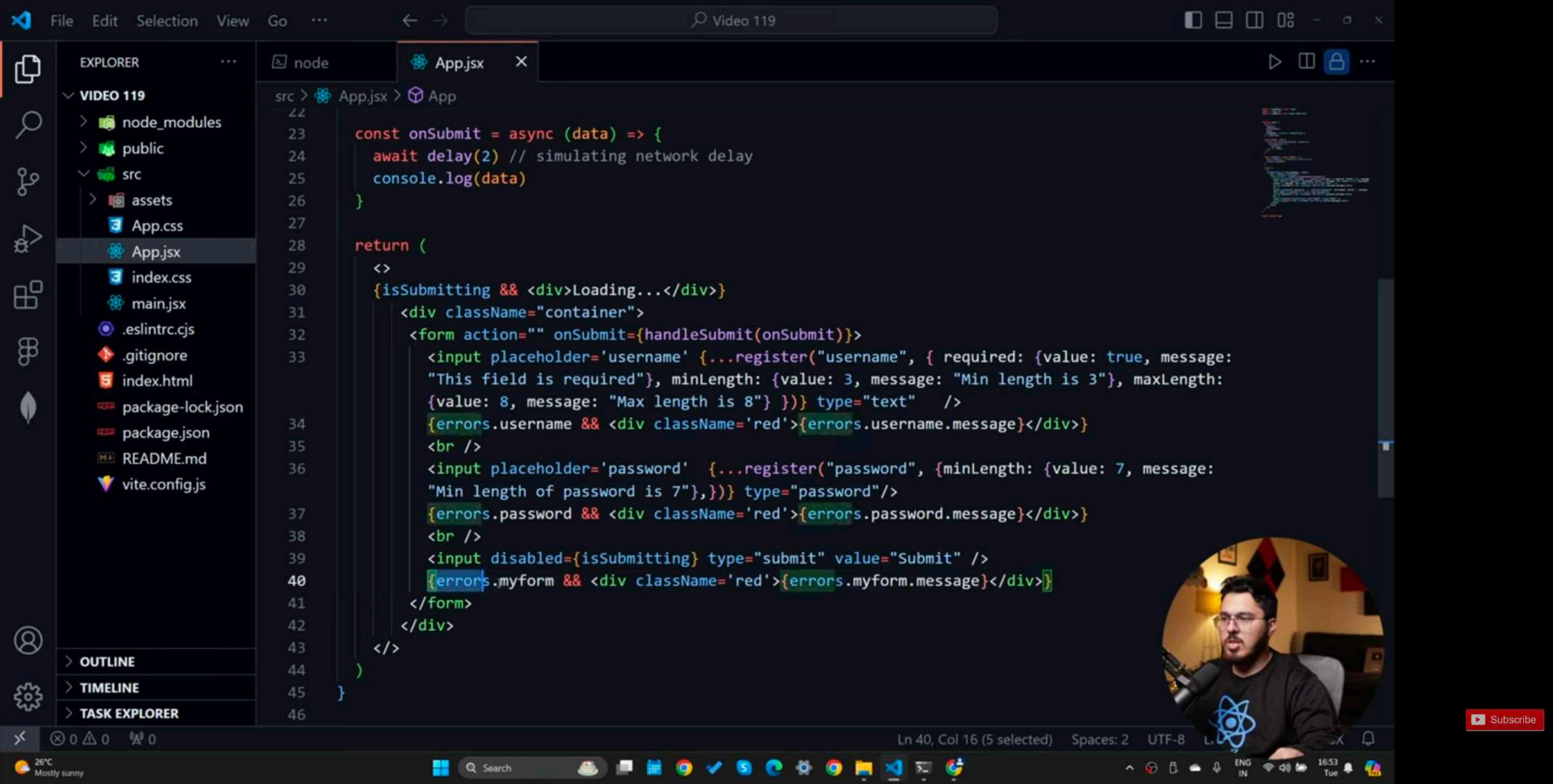
Ln 30, Col 37 Spaces: 2 UTF-8 L 16:46 Tue ENG IN

26°C Mostly sunny

Search

Subscribe





File Edit Selection View Go ... ← → Video 119 □ □ □ 00 - o x □ □ □ ...

EXPLORER ...

VIDEO 119

- > node_modules
- > public
- src
- > assets
- App.css
- App.jsx
- index.css
- main.jsx
- .eslintrc.cjs
- .gitignore
- index.html
- package-lock.json
- package.json
- README.md
- vite.config.js

OUTLINE

TIMELINE

TASK EXPLORER

node App.jsx

```
src > App.jsx > App > setError
1 import { useState } from 'react'
2 import './App.css'
3 import { useForm } from "react-hook-form"
4
5
6
7 function App() {
8   const [
9     register,
10    handleSubmit,
11    setError,
12    formState: { errors, isSubmitting },
13  ] = useForm();
14
15  const delay = (d)=>{
16    return new Promise((resolve, reject)=>{
17      setTimeout(() => {
18        resolve()
19      }, d * 1000);
20    })
21  }
22
23  const onSubmit = async (data) => {
24    await delay(2) // simulating network delay
25    console.log(data)
26  }
27
28  return (

```

Ln 11, Col 13 Spaces: 2 UTF-8 L 26°C Mostly sunny

26°C Mostly sunny

Search

16:53 Tue ENG IN

Subscribe



File Edit Selection View Go ... ← → Video 119 □ □ □ 00 - o x □ □ □ 000 ...

EXPLORER ...

VIDEO 119

- > node_modules
- > public
- src
 - > assets
 - App.css
 - App.jsx
 - index.css
 - main.jsx
 - .eslintrc.cjs
 - .gitignore
 - index.html
 - package-lock.json
 - package.json
 - README.md
 - vite.config.js

OUTLINE

TIMELINE

TASK EXPLORER

node App.jsx

```
src > App.jsx > App > [o] onSubmit
10   handleSubmit,
11   setError,
12   formState: { errors, isSubmitting },
13 } = useForm();

14
15 const delay = (d)=>{
16   return new Promise((resolve, reject)=>{
17     setTimeout(() => {
18       resolve()
19     }, d * 1000);
20   })
21 }

22
23 const onSubmit = async (data) => {
24   await delay(2) // simulating network delay
25   console.log(data)
26   if(data.username !== "shubham"){
27     setError("myform")
28   }
29 }

30
31 return (
32   <>
33   {isSubmitting && <div>Loading...</div>}
34   <div className="container">
35     <form action="" onSubmit={handleSubmit(onSubmit)}>
36       <input placeholder='username' {...register("username", { required: { value: true, message: "This field is required" }, minLength: { value: 3, message: "Min length is 3" }, maxLength: 10 })}>
```

Ln 27, Col 23 (6 selected) Spaces: 2 UTF-8

26°C Mostly sunny

Search

16:53 Tue ENG IN



Subscribe

File Edit Selection View Go ... ← → Video 119 □ □ □ □ - □ ×

EXPLORER ... node App.jsx X

VIDEO 119

- > node_modules
- > public
- src
 - > assets
 - App.css
 - App.jsx
 - index.css
 - main.jsx
 - .eslintrc.cjs
 - .gitignore
 - index.html
 - package-lock.json
 - package.json
 - README.md
 - vite.config.js

OUTLINE

TIMELINE

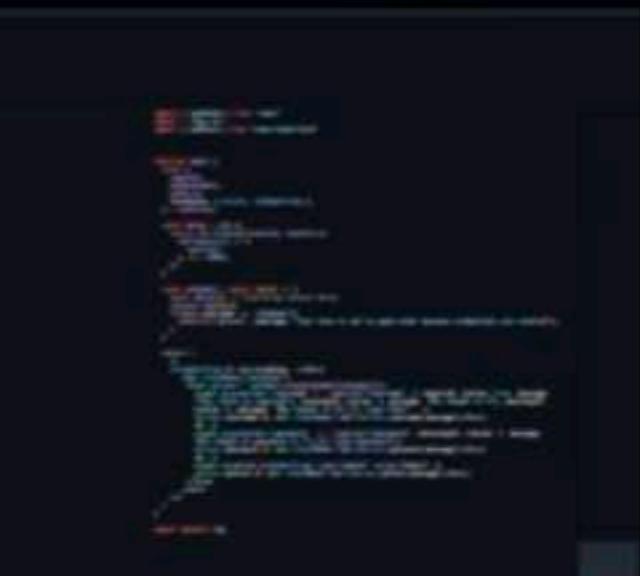
TASK EXPLORER

App.jsx

```
src > App.jsx > App > onSubmit
16   return new Promise((resolve, reject)=>{
17     setTimeout(() => {
18       resolve()
19     }, d * 1000);
20   })
21
22
23 const onSubmit = async (data) => {
24   await delay(2) // simulating network delay
25   console.log(data)
26   if(data.username !== "shubham"){
27     setError("myform", {message: "Your form is not in good order because credentials are invalid"})
28   }
29 }
30
31 return (
32   <>
33   {isSubmitting && <div>Loading...</div>}
34   <div className="container">
35     <form action="" onSubmit={handleSubmit(onSubmit)}>
36       <input placeholder='username' {...register("username", { required: {value: true, message: "This field is required"}, minLength: {value: 3, message: "Min length is 3"}, maxLength: {value: 8, message: "Max length is 8"} })} type="text" />
37       {errors.username && <div className='red'>{errors.username.message}</div>}
38       <br />
39       <input placeholder='password' {...register("password", {minLength: {value: 7, message: "Min length of password is 7"},}}) type="password"/>
40       {errors.password && <div className='red'>{errors.password.message}</div>}
```

Ln 27, Col 101 Spaces: 2 UTF-8 L. 26°C Mostly sunny

Search



Subscribe



App.jsx > App > [o] onSubmit > message

```
16     return new Promise((resolve, reject) => ▶▶▶
17       setTimeout(() => {
18         resolve()
19       }, d * 1000);
20     })
21   }
22
23   const onSubmit = async (data) => {
24     await delay(2) // simulating network delay
25     console.log(data)
26     if(data.username !== "shubham"){
27       setError("myform", {message: "Your form is not in good order because credentials are invalid"})
28     }
29     else if(data.username === "rohan"){
30       setError("blocked", {message: "Sorry this user is blocked"})
31     }
32   }
33
34   return (
35     <>
36     {isSubmitting && <div>Loading...</div>}
37     <div className="container">
38       <form action="" onSubmit={handleSubmit(onSubmit)}>
39         <input placeholder='username' {...register("username", { required: {value: true, r
40           "This field is required"}, minLength: {value: 3, message: "Min length is 3"}, max
41           {value: 8, message: "Max length is 8"} })} type="text" />
42         {errors.username && <div className='red'>{errors.username.message}</div>}
43         <br />
```



File Edit Selection View Go ... ← → Video 119 □ □ □ 00 - o x

EXPLORER ... node App.jsx X

VIDEO 119

node_modules

public

src

assets

App.css

App.jsx

index.css

main.jsx

.eslintrc.cjs

.gitignore

index.html

package-lock.json

package.json

README.md

vite.config.js

OUTLINE

TIMELINE

TASK EXPLORER

2x ►►

```
33
34     return (
35         <>
36         {isSubmitting && <div>Loading...</div>}
37         <div className="container">
38             <form action="" onSubmit={handleSubmit(onSubmit)}>
39                 <input placeholder='username' {...register("username", { required: {value: true, message: "This field is required"}, minLength: {value: 3, message: "Min length is 3"}, maxLength: {value: 8, message: "Max length is 8"} })} type="text" />
40                 {errors.username && <div className='red'>{errors.username.message}</div>}
41                 <br />
42                 <input placeholder='password' {...register("password", {minLength: {value: 7, message: "Min length of password is 7"} })} type="password"/>
43                 {errors.password && <div className='red'>{errors.password.message}</div>}
44                 <br />
45                 <input disabled={isSubmitting} type="submit" value="Submit" />
46                 {errors.myform && <div className='red'>{errors.myform.message}</div>}
47                 {errors.blocked && <div className='red'>{errors.blocked.message}</div>}
48             </form>
49         </div>
50     </>
51 )
52 }
53
54 export default App
55
```

Ln 47, Col 66 Spaces: 2 UTF-8 L ENG IN 16:57 Tue

26°C Mostly sunny

Search

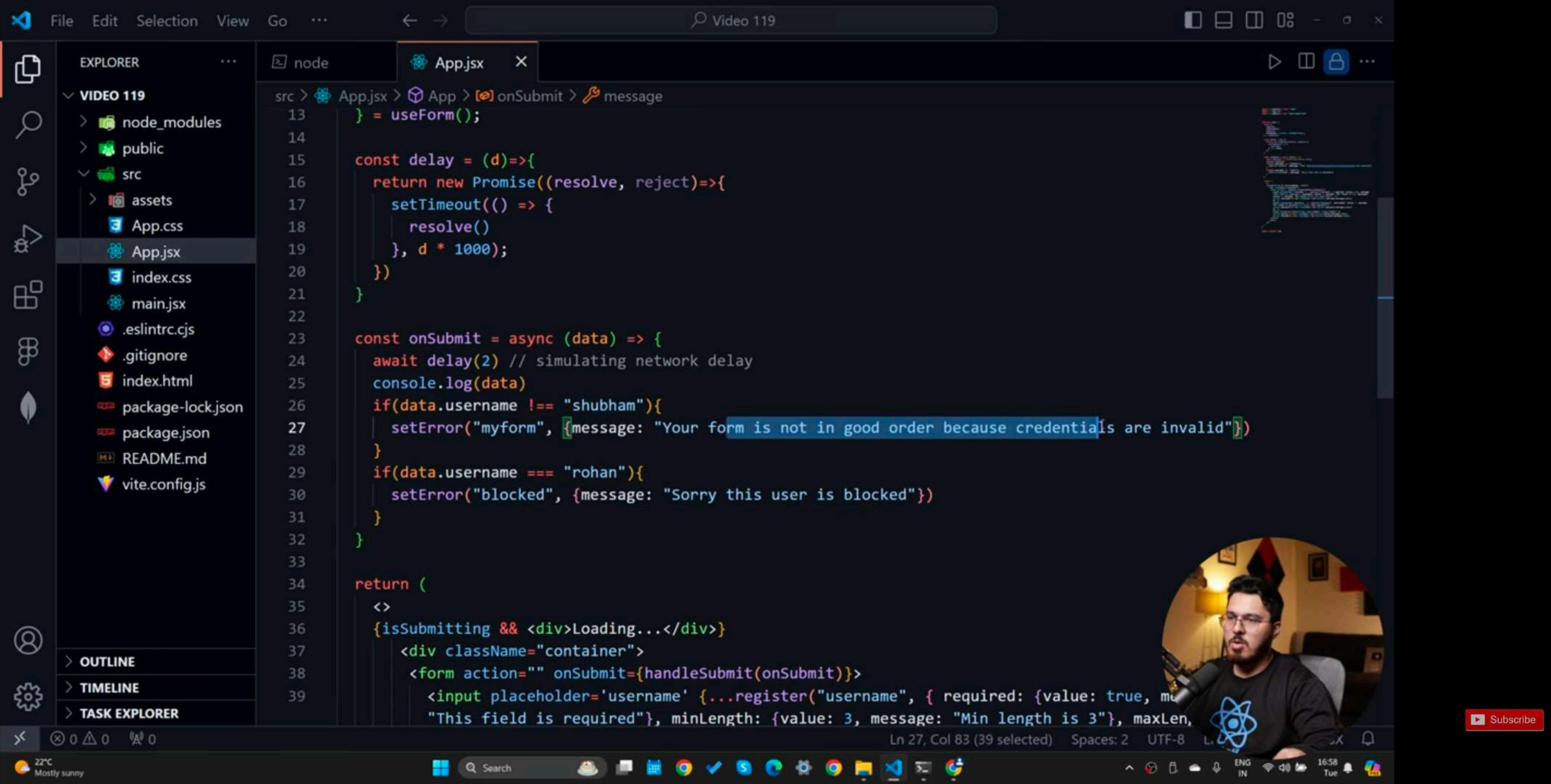
Subscriptions

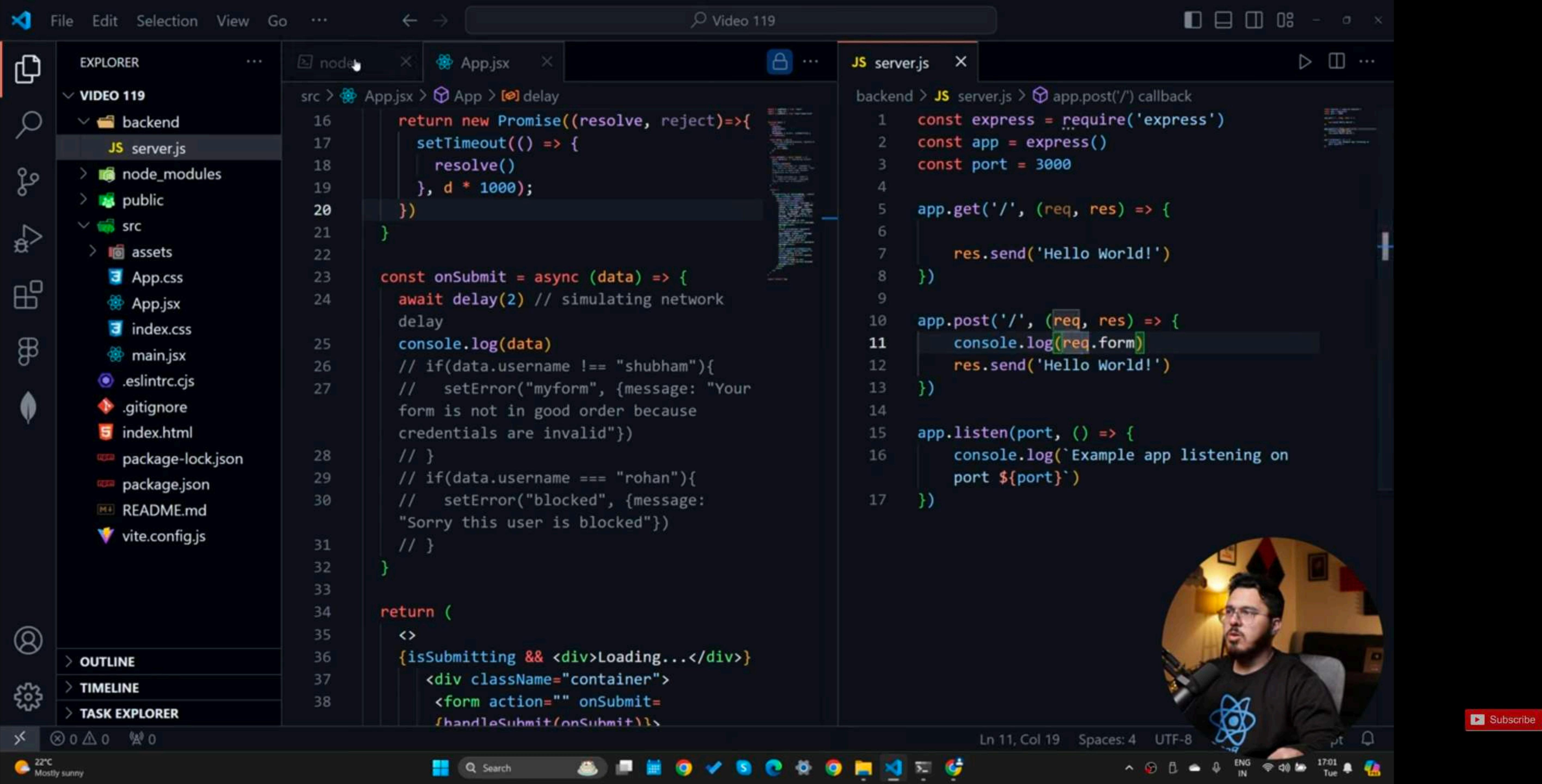
Notifications

Atom icon

Subscribe







5:24 (2)

3.47
LTE KB/s 5G 83%



Your like can help me bring more awesome courses like this!

Learn Redux in One video | Sigma Web Development Course - Tutorial #120

129K views 1y ago Sigma Web Development Course - Web ...more



CodeWithHarry 7.98M

Join



4.1K | 1.1K

Share

Remix

Download

Comments 275



Add a comment...



End of playlist



Today's Special • 2/2



Sigma Web Development Course - VI | redux - Google Search | Redux - A predictable state container...

redux.js.org

Sigma Web Developme... GitHub - CodeWithHarr... Learn to code online - C... Es with Express

Redux

Getting Started Tutorial Usage Guide API FAQ Best Practices GitHub Need help? ⚙️

Search

Redux

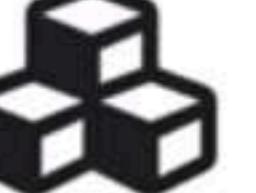
A Predictable State Container for JS Apps

Get Started



Predictable

Redux helps you write applications that **behave consistently**, run in different environments (client, server, and native), and are **easy to test**.



Centralized

Centralizing your application's state and logic enables powerful capabilities like **undo/redo**, **state persistence**, and much more.



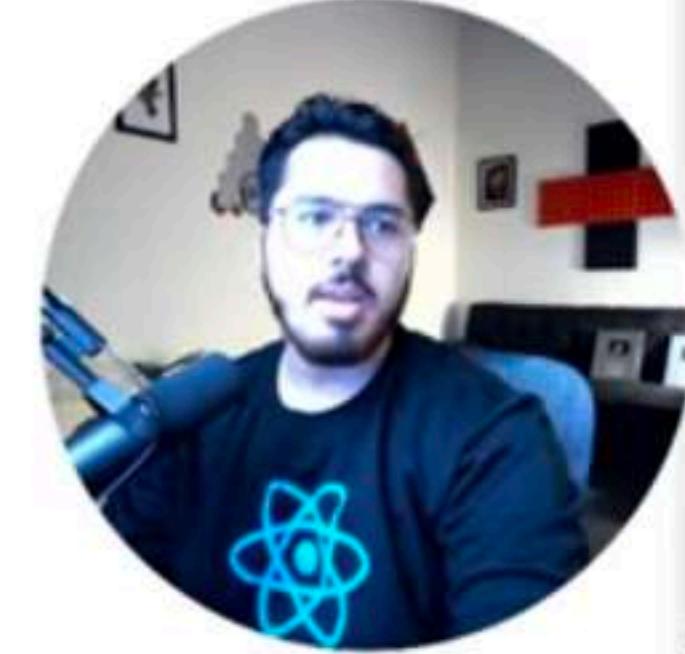
Debuggable

The Redux DevTools make it easy to trace **when, where, why, and how your application's state changed**. Redux's architecture lets you log changes, use **"time-travel debugging"**, and even send complete error reports to a server.



Flexible

Redux **works with any UI layer**, and has a **large ecosystem of addons** to fit your needs.



Other Libraries from the Redux Team

React-Redux ↗

Redux Toolkit ↗

Subscribe

24°C Mostly cloudy

Search

17:36 Wed

Sigma Web Development Course - VI | redux - Google Search | Redux Toolkit | Redux Toolkit

redux-toolkit.js.org

Sigma Web Developme... GitHub - CodeWithHarry... Learn to code online - C... Eds with Express

Redux Toolkit

Getting Started Tutorials Usage Guide API RTK Query GitHub

Search

Redux Toolkit

The official, opinionated, batteries-included toolset for efficient Redux development

Get Started



Simple

Includes utilities to simplify common use cases like **store setup**, **creating reducers**, **immutable update logic**, and more.



Opinionated

Provides **good defaults for store setup out of the box**, and includes the **most commonly used Redux addons** built-in.



Powerful

Takes inspiration from libraries like Immer and Autodux to let you **write "mutative" immutable update logic**, and even **create entire "slices" of state automatically**.



Effective

Lets you focus on the core logic your app needs, so you can **do more work with less code**.



Other Libraries from the Redux Team

Subscribe

Sigma Web Development Course - VI Quick Start | Redux Toolkit

redux-toolkit.js.org/tutorials/quick-start

Sigma Web Developme... GitHub - CodeWithHarr... Learn to code online - C... Es with Express

Redux Toolkit Quick Start

WHAT YOU'LL LEARN

- How to set up and use Redux Toolkit with React-Redux

PREREQUISITES

- Familiarity with [ES6 syntax and features](#)
- Knowledge of React terminology: [JSX](#), [State](#), [Function Components](#), [Props](#), and [Hooks](#)
- Understanding of [Redux terms and concepts](#)

Introduction

Welcome to the Redux Toolkit Quick Start tutorial! This tutorial will briefly introduce you to Redux Toolkit and teach you how to start using it correctly.

How to Read This Tutorial

This page will focus on just how to set up a Redux application with Redux Toolkit and the main APIs you'll use. For explanations of what Redux is, how it works, and full examples of how to use Redux Toolkit, [see the tutorials linked in the "Tutorials Overview" page](#).

For this tutorial, we assume that you're using Redux Toolkit with React, but you can also use it with other UI layers as well. The examples are based on a typical [Create-React-App folder structure](#) where all the application code is in a `src`, but the patterns can be adapted to whatever project or folder setup you're using.

The [Redux+JS template for Create-React-App](#) comes with this same project setup already configured.



Subscribe

Sigma Web Development Course - V... Quick Start | Redux Toolkit

redux-toolkit.js.org/tutorials/quick-start

Sigma Web Developme... GitHub - CodeWithHarr... Learn to code online - C... EJS with Express

Redux Toolkit

Getting Started Tutorials Usage Guide API RTK Query GitHub

Search

Introduction

- Getting Started
- Why Redux Toolkit is How To Use Redux Today

Tutorials

- Tutorials Overview
- Quick Start
- TypeScript Quick Start
- RTK Query Quick Start

Using Redux Toolkit

- Migrations
- Usage Guide
- Usage With TypeScript
- Writing Reducers with Immer
- Setup with Next.js

API Reference

RTK Query

- RTK Query Overview
- Comparison with Other Tools
- Examples
- Usage With TypeScript

Using RTK Query

API Reference

Install Redux Toolkit and React-Redux

Add the Redux Toolkit and React-Redux packages to your project:

```
npm install @reduxjs/toolkit react-redux
```

Create a Redux Store

Create a file named `src/app/store.js`. Import the `configureStore` API from Redux Toolkit. We'll start by creating an empty Redux store, and exporting it:

TypeScript JavaScript

```
app/store.js
```

```
import { configureStore } from '@reduxjs/toolkit'

export const store = configureStore({
  reducer: {}
})

// Infer the 'RootState' and 'AppDispatch' types from the store itself
export type RootState = ReturnType<typeof store.getState>
// Inferred type: {posts: PostsState, comments: CommentsState, users: UsersState}
export type AppDispatch = typeof store.dispatch
```

This creates a Redux store, and also automatically configure the Redux DevTools extension so that you can inspect the store while developing.

Provide the Redux Store to React

Once the store is created, we can make it available to our React components by putting a React-Redux `<Provider>` around our application in `src/index.js`. Import the Redux store we just created, put a `<Provider>` around your `<App>`, and pass



Subscribe

File Edit Selection View Go ... ← → Video 120 □ □ □ 00 - □ ×

EXPLORER

VIDEO 120

- > node_modules
- > public
- src
- > assets
- App.css
- App.jsx
- index.css
- main.jsx
- .eslintrc.cjs
- .gitignore
- index.html
- package-lock.json
- package.json
- README.md
- vite.config.js

powershell X

✓ Select a variant: » **JavaScript**

Scaffolding project in C:\Users\iitia\Downloads\Sigma Web Development Course\Sigma-Web-Dev-Course\Video 120...

Done. Now run:

```
npm install
npm run dev
```

PS C:\Users\iitia\Downloads\Sigma Web Development Course\Sigma-Web-Dev-Course\Video 120> **npm i**

added 273 packages, and audited 274 packages in 10s

98 packages are looking for funding
run `npm fund` for details

found 0 vulnerabilities

PS C:\Users\iitia\Downloads\Sigma Web Development Course\Sigma-Web-Dev-Course\Video 120> **npm install @reduxjs/toolkit react-redux**

[.....] - idealTree:Video 120: sill idealTree buildDeps

OUTLINE

TIMELINE

TASK EXPLORER

24°C Mostly cloudy

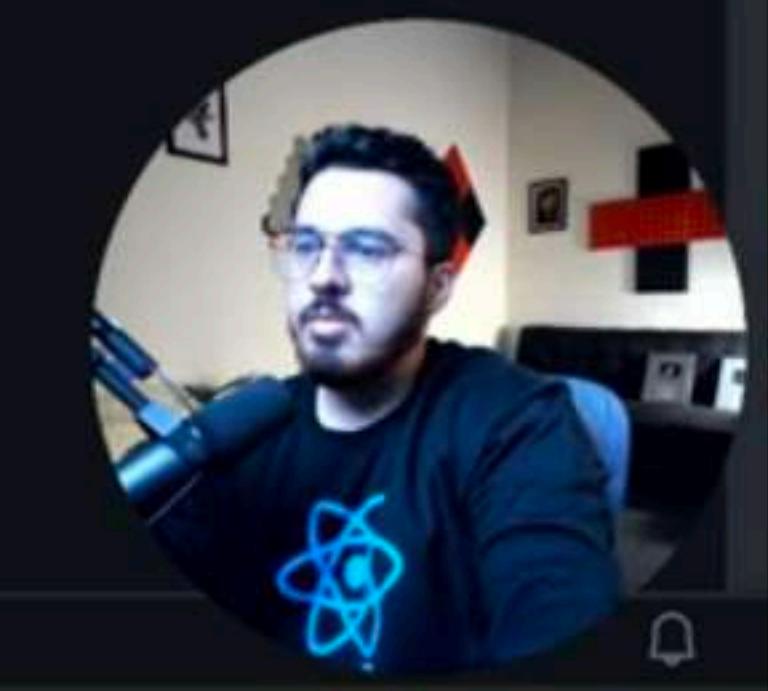
Search

24°C Mostly cloudy

ENG IN

17:38 Wed

Subscribe



File Edit Selection View Go ... ⟲ ⟳ Video 120 ⟳ + 🔍 ...

EXPLORER

VIDEO 120

- > node_modules
- > public
- src

 - > assets
 - App.css
 - App.jsx
 - index.css
 - main.jsx
 - .eslintrc.cjs
 - .gitignore
 - index.html
 - package-lock.json
 - package.json
 - README.md
 - vite.config.js

powershell X

✓ Select a variant: » JavaScript

Scaffolding project in C:\Users\iitia\Downloads\Sigma Web Development Course\Sigma-Web-Dev-Course\Video 120...

Done. Now run:

```
npm install
npm run dev
```

PS C:\Users\iitia\Downloads\Sigma Web Development Course\Sigma-Web-Dev-Course\Video 120> npm i

added 273 packages, and audited 274 packages in 10s

98 packages are looking for funding
run `npm fund` for details

found 0 vulnerabilities

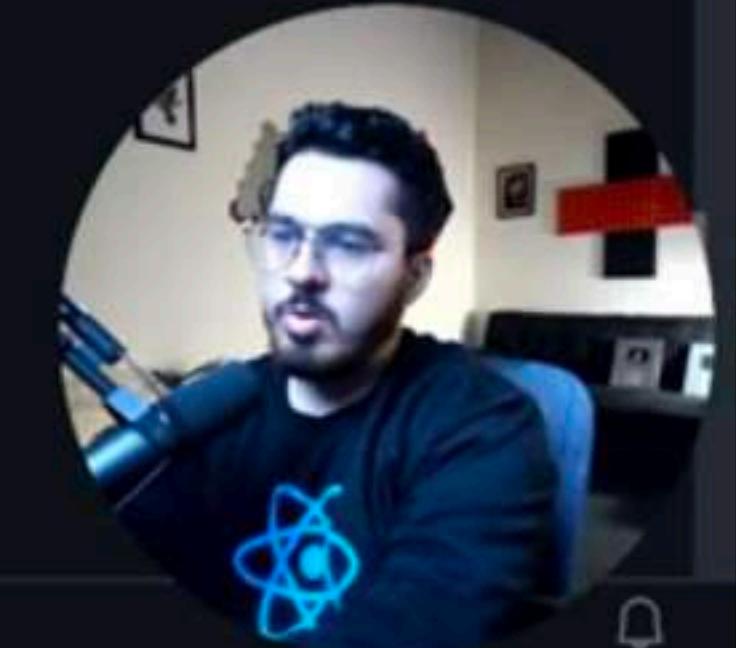
PS C:\Users\iitia\Downloads\Sigma Web Development Course\Sigma-Web-Dev-Course\Video 120> npm install @reduxjs/toolkit react-redux

added 8 packages, and audited 282 packages in 2s

99 packages are looking for funding
run `npm fund` for details

found 0 vulnerabilities

PS C:\Users\iitia\Downloads\Sigma Web Development Course\Sigma-Web-Dev-Course\Video 120> npm run dev



24°C Mostly cloudy

Search

ENG IN 17:38 Wed

Subscribe

File Edit Selection View Go ... ← → Video 120

EXPLORER

VIDEO 120

- > node_modules
- > public
- src
 - > assets
 - components
 - Navbar.jsx
 - App.css
 - App.jsx
 - index.css
 - main.jsx
- .eslintrc.cjs
- .gitignore
- index.html
- package-lock.json
- package.json
- README.md
- vite.config.js

OUTLINE

TIMELINE

TASK EXPLORER

node App.jsx

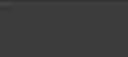
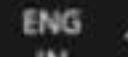
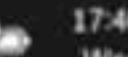
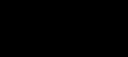
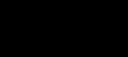
```
1 import { useState } from 'react'
2 import reactLogo from './assets/react.svg'
3 import viteLogo from '/vite.svg'
4 import './App.css'
5 import Navbar from './components/Navbar'
6
7 function App() {
8   const [count, setCount] = useState(0)
9
10  return (
11    <>
12    <Navbar count={count}>
13      <div>
14        <a href="https://vitejs.dev" target="_blank">
15          <img src={viteLogo} className="logo" alt="Vite logo" />
16        </a>
17        <a href="https://react.dev" target="_blank">
18          <img src={reactLogo} className="logo react" alt="React logo" />
19        </a>
20      </div>
21      <h1>Vite + React</h1>
22      <div className="card">
```

Navbar.jsx

```
1 import React from 'react'
2
3 const Navbar = ({count}) => {
4   return (
5     <div>
6       I am a navbar and counter is {count}
7     </div>
8   )
9 }
10
11 export default Navbar
```

Ln 6, Col 42 Spaces: 2 UTF-8 CRLF { }  

24°C Mostly cloudy

Search                ENG IN 17:40 Wed

File Edit Selection View Go ... ← → Video 120

EXPLORER

VIDEO 120

- > node_modules
- > public
- src
 - > assets
 - > components
 - redux
 - JS store.js
 - App.css
 - App.jsx
 - index.css
 - main.jsx
 - .eslintrc.cjs
 - .gitignore
 - index.html
 - package-lock.json
 - package.json
 - README.md
 - vite.config.js

OUTLINE

TIMELINE

TASK EXPLORER

node App.jsx

JS store.js

```
src > App.jsx > App
1 import { useState } from 'react'
2 import reactLogo from './assets/react.svg'
3 import viteLogo from '/vite.svg'
4 import './App.css'
5 import Navbar from './components/Navbar'

6
7 function App() {
8   const [count, setCount] = useState(0)

9
10  return (
11    <>
12      <Navbar/>
13      <div>
14        <a href="https://vitejs.dev"
15          target="_blank">
16          <img src={viteLogo}
17            className="logo" alt="Vite logo" />
18        </a>
19        <a href="https://react.dev"
20          target="_blank">
21          <img src={reactLogo}
22            className="logo react" alt="React logo" />
23        </a>
24      </div>
25      <h1>Vite + React</h1>
26      <div className="card">
```

Ln 1, Col 1 Spaces: 4 UTF-8 CRLF

24°C Mostly cloudy

Search

17:42 Wed

ENG IN

Subscribe



Sigma Web Development Course - VI Quick Start | Redux Toolkit Vite + Read

redux-toolkit.js.org/tutorials/quick-start

Sigma Web Developme... GitHub - CodeWithHarr... Learn to code online - C... Es with Express

Redux Toolkit

Getting Started Tutorials Usage Guide API RTK Query GitHub

Search

Introduction

- Getting Started
- Why Redux Toolkit is How To Use Redux Today

Tutorials

- Tutorials Overview
- Quick Start
- TypeScript Quick Start
- RTK Query Quick Start

Using Redux Toolkit

- Migrations
- Usage Guide
- Usage With TypeScript
- Writing Reducers with Immer
- Setup with Next.js

API Reference

- RTK Query
- RTK Query Overview
- Comparison with Other Tools
- Examples
- Usage With TypeScript
- Using RTK Query
- API Reference

Create a Redux Store

Create a file named `src/app/store.js`. Import the `configureStore` API from Redux Toolkit. We'll start by creating an empty Redux store, and exporting it:

TypeScript [JavaScript](#)

```
app/store.js
import { configureStore } from '@reduxjs/toolkit'

export const store = configureStore({
  reducer: {}
})
```

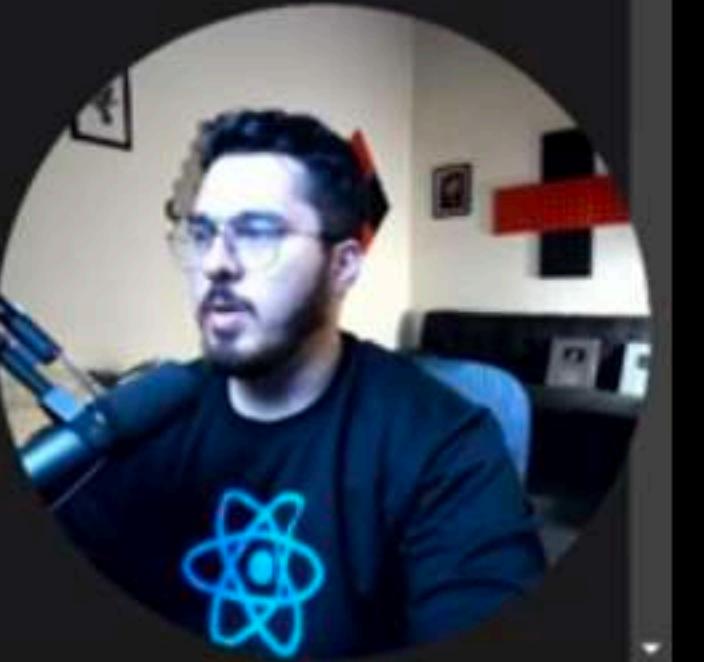
This creates a Redux store, and also automatically configure the Redux DevTools extension so that you can inspect the store while developing.

Provide the Redux Store to React

Once the store is created, we can make it available to our React components by putting a React-Redux `<Provider>` around our application in `src/index.js`. Import the Redux store we just created, put a `<Provider>` around your `<App>`, and pass the store as a prop:

TypeScript [JavaScript](#)

```
index.js
import React from 'react'
import ReactDOM from 'react-dom'
import './index.css'
import App from './App'
import { store } from './app/store'
import { Provider } from 'react-redux'
```



Subscribe

File Edit Selection View Go ... ← → Video 120

EXPLORER

VIDEO 120

- > node_modules
- > public
- src
 - > assets
 - > components
 - redux
 - JS store.js
 - App.css
 - App.jsx
 - index.css
 - main.jsx
 - .eslintrc.cjs
 - .gitignore
 - index.html
 - package-lock.json
 - package.json
 - README.md
 - vite.config.js

node App.jsx JS store.js

src > redux > JS store.js > [x] store

```
1 import { configureStore } from '@reduxjs/toolkit'
2
3 export const store = configureStore({
4   reducer: {},
5 })
```

Ln 5, Col 3 Spaces: 4 UTF-8 CRLF

24°C Mostly cloudy

Search

Subscribe



Sigma Web Development Course - V... Quick Start | Redux Toolkit Vite + React redux-toolkit.js.org/tutorials/quick-start Sigma Web Developme... GitHub - CodeWithHarr... Learn to code online - C... Ejs with Express

Redux Toolkit

Introduction

Getting Started

Why Redux Toolkit is How To Use Redux Today

Tutorials

Tutorials Overview

Quick Start

TypeScript Quick Start

RTK Query Quick Start

Using Redux Toolkit

Migrations

Usage Guide

Usage With TypeScript

Writing Reducers with Immer

Setup with Next.js

API Reference

RTK Query

RTK Query Overview

Comparison with Other Tools

Examples

Usage With TypeScript

Using RTK Query

API Reference

Getting Started

Tutorials

Usage Guide

API

RTK Query

GitHub

Search

Introduction

How to Read This Tutorial

Usage Summary

Install Redux Toolkit and React-Redux

Create a Redux Store

Provide the Redux Store to React

Create a Redux State Slice

Add Slice Reducers to the Store

Use Redux State and Actions in React Components

What You've Learned

Full Counter App Example

What's Next?

index.js

```
import React from 'react'
import ReactDOM from 'react-dom'
import './index.css'
import App from './App'

import { store } from './app/store'
import { Provider } from 'react-redux'

ReactDOM.render(
  <Provider store={store}>
    <App />
  </Provider>,
  document.getElementById('root')
)
```

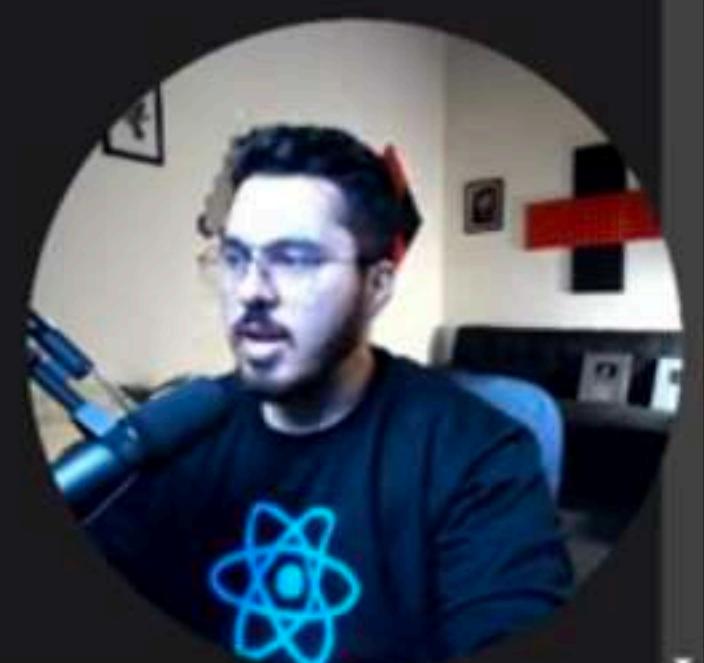
Provide the Redux Store to React

Once the store is created, we can make it available to our React components by putting a React-Redux `<Provider>` around our application in `src/index.js`. Import the Redux store we just created, put a `<Provider>` around your `<App>`, and pass the store as a prop:

TypeScript [JavaScript](#)

Create a Redux State Slice

Add a new file named `src/features/counter/counterSlice.js`. In that file, import the `createSlice` API from Redux Toolkit



24°C Mostly cloudy

Search

Clouds

Temperature

Wind

Humidity

UV Index

17:42

Wednesday

ENG IN

File Edit Selection View Go ... ← → Video 120

EXPLORER

VIDEO 120

- > node_modules
- > public
- > src
 - > assets
 - > components
 - > redux
 - JS store.js
 - App.css
 - App.jsx
 - index.css
 - main.jsx
 - > .eslintrc.cjs
 - > .gitignore
 - > index.html
 - > package-lock.json
 - > package.json
 - > README.md
 - > vite.config.js

node App.jsx store.js main.jsx

src > main.jsx

```
1 import React from 'react'
2 import ReactDOM from 'react-dom/client'
3 import App from './App.jsx'
4 import './index.css'
5 import {store} from './redux/store.js'

6
7 ReactDOM.createRoot(document.getElementById('root')).render(
8   <React.StrictMode>
9     <App />
10    </React.StrictMode>,
11  )
12
```

Ln 5, Col 38 Spaces: 2 UTF-8 LF { } 24°C Mostly cloudy

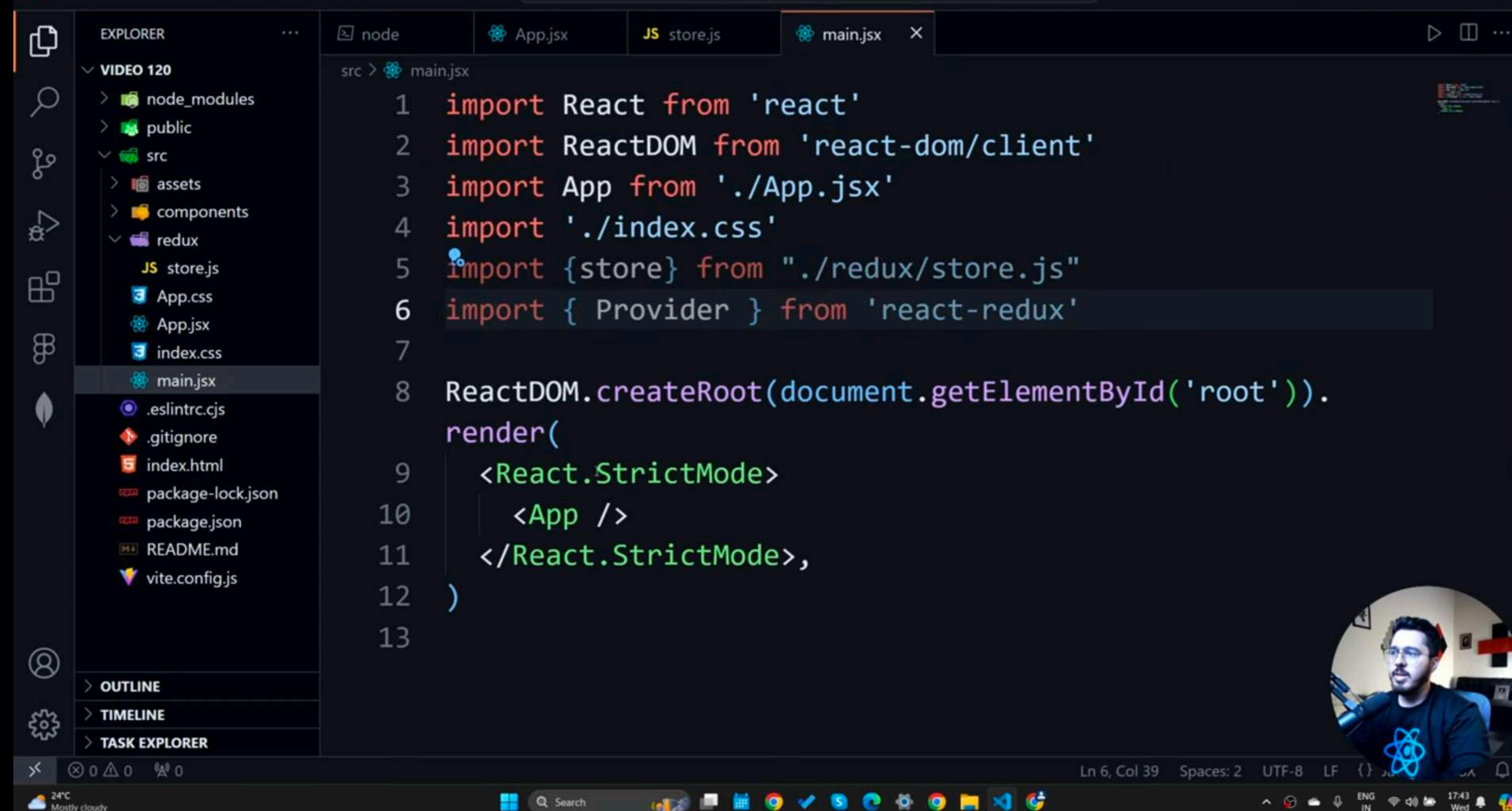
Search

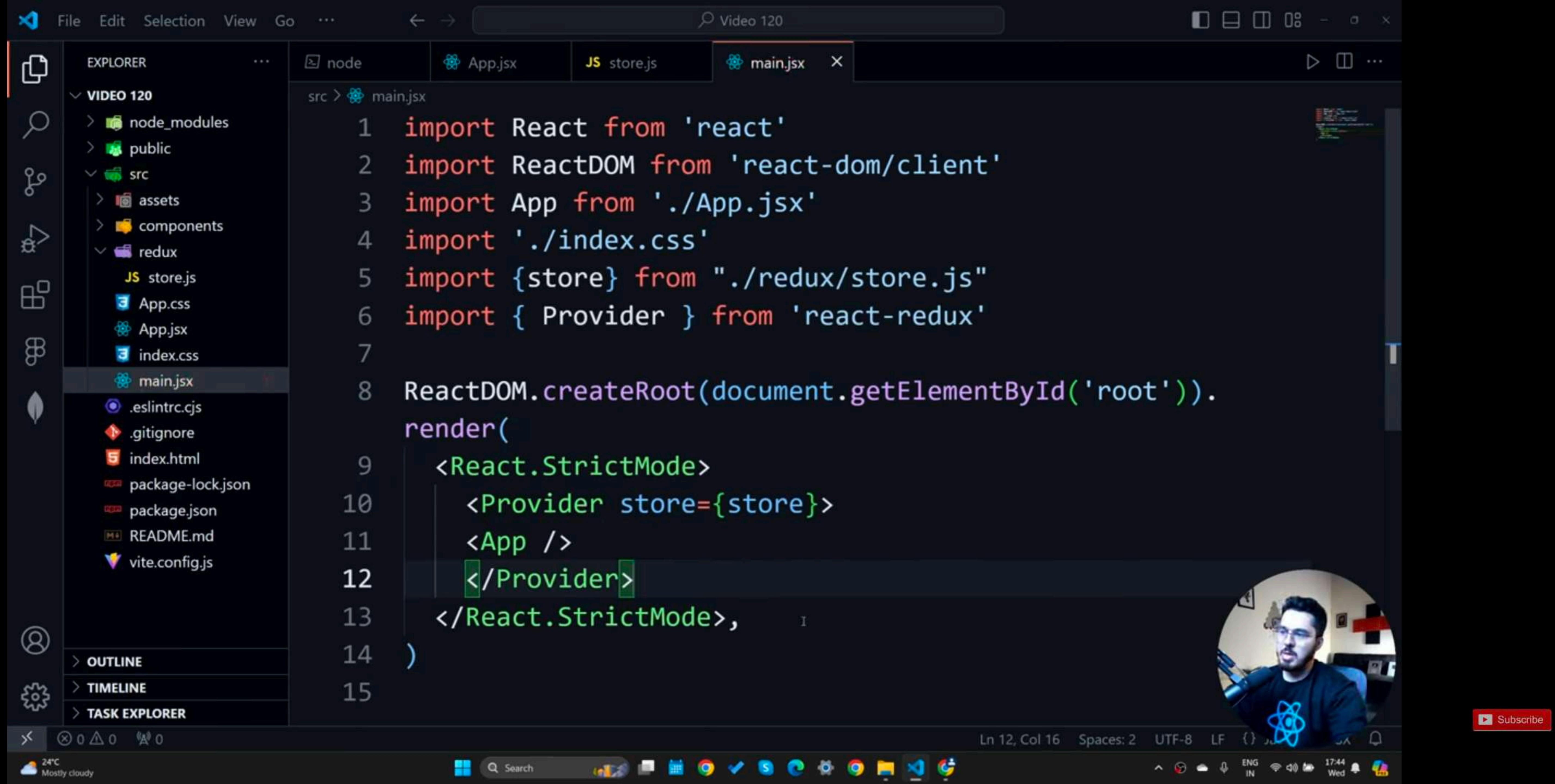
24°C Mostly cloudy

17:43 Wed ENG IN

Subscribe







Sigma Web Development Course - VI Quick Start | Redux Toolkit Vite + React redux-toolkit.js.org/tutorials/quick-start Sigma Web Developme... GitHub - CodeWithHarr... Learn to code online - C... EJS with Express

Redux Toolkit

Getting Started Tutorials Usage Guide API RTK Query GitHub

Introduction

Getting Started

Why Redux Toolkit is How To Use Redux Today

Tutorials

Tutorials Overview

Quick Start

TypeScript Quick Start

RTK Query Quick Start

Using Redux Toolkit

Migrations

Usage Guide

Usage With TypeScript

Writing Reducers with Immer

Setup with Next.js

API Reference

RTK Query

RTK Query Overview

Comparison with Other Tools

Examples

Usage With TypeScript

Using RTK Query

API Reference

Create a Redux State Slice

Add a new file named `src/features/counter/counterSlice.js`. In that file, import the `createSlice` API from Redux Toolkit.

Creating a slice requires a string name to identify the slice, an initial state value, and one or more reducer functions to define how the state can be updated. Once a slice is created, we can export the generated Redux action creators and the reducer function for the whole slice.

Redux requires that we write all state updates immutably, by making copies of data and updating the copies. However, Redux Toolkit's `createSlice` and `createReducer` APIs use `immer` inside to allow us to write "mutating" update logic that becomes correct immutable updates.

TypeScript [JavaScript](#)

```
features/counter/counterSlice.js
```

```
import { createSlice } from '@reduxjs/toolkit'

const initialState = {
  value: 0,
}

export const counterSlice = createSlice({
  name: 'counter',
  initialState,
  reducers: {
    increment: (state) => {
```

Introduction

How to Read This Tutorial

Usage Summary

Install Redux Toolkit and React-Redux

Create a Redux Store

Provide the Redux Store to React

Create a Redux State Slice

Add Slice Reducers to the Store

Use Redux State and Actions in React Components

What You've Learned

Full Counter App Example

What's Next?



Subscribe

Sigma Web Development Course - V... X Quick Start | Redux Toolkit X Vite + React X + redux-toolkit.js.org/tutorials/quick-start

Sigma Web Developme... GitHub - CodeWithHarr... Learn to code online - C... Ejs with Express

Redux Toolkit

Introduction

- Getting Started
- Why Redux Toolkit is How To Use Redux Today

Tutorials

- Tutorials Overview
- Quick Start
- TypeScript Quick Start
- RTK Query Quick Start

Using Redux Toolkit

- Migrations
- Usage Guide
- Usage With TypeScript
- Writing Reducers with Immer
- Setup with Next.js

API Reference

RTK Query

- RTK Query Overview
- Comparison with Other Tools
- Examples
- Usage With TypeScript
- Using RTK Query
- API Reference

Toolkit.

Creating a slice requires a string name to identify the slice, an initial state value, and one or more reducer functions to define how the state can be updated. Once a slice is created, we can export the generated Redux action creators and the reducer function for the whole slice.

Redux requires that we write all state updates immutably, by making copies of data and updating the copies. However, Redux Toolkit's `createSlice` and `createReducer` APIs use `Immer` inside to allow us to write "mutating" update logic that becomes correct immutable updates.

TypeScript [JavaScript](#)

```
features/counter/counterSlice.js
```

```
import { createSlice } from '@reduxjs/toolkit'

const initialState = {
  value: 0,
}

export const counterSlice = createSlice({
  name: 'counter',
  initialState,
  reducers: {
    increment: (state) => {
      // Redux Toolkit allows us to write "mutating" logic in reducers. It
      // doesn't actually mutate the state because it uses the Immer library,
      // which detects changes to a "draft state" and produces a brand new
      // immutable state based off those changes
      state.value += 1
    },
    decrement: (state) => {
      state.value -= 1
    },
    incrementByAmount: (state, action) => {
      state.value += action.payload
    },
  },
})
```

Introduction

- How to Read This Tutorial
- Usage Summary
- Install Redux Toolkit and React-Redux
- Create a Redux Store
- Provide the Redux Store to React
- Create a Redux State Slice
- Add Slice Reducers to the Store
- Use Redux State and Actions in React Components

What You've Learned

- Full Counter App Example

What's Next?



Subscribe

File Edit Selection View Go ... ← → Video 120

node App.jsx store.js main.jsx counterSlice.js

EXPLORER ...

VIDEO 120 node App.jsx store.js main.jsx counterSlice.js

src > redux > counter > JS counterSlice.js

1

node_modules

public

src

assets

components

redux

counter

JS counterSlice.js

JS store.js

App.css

App.jsx

index.css

main.jsx

.eslintrc.cjs

.gitignore

index.html

package-lock.json

package.json

README.md

vite.config.js

OUTLINE

TIMELINE

TASK EXPLORER

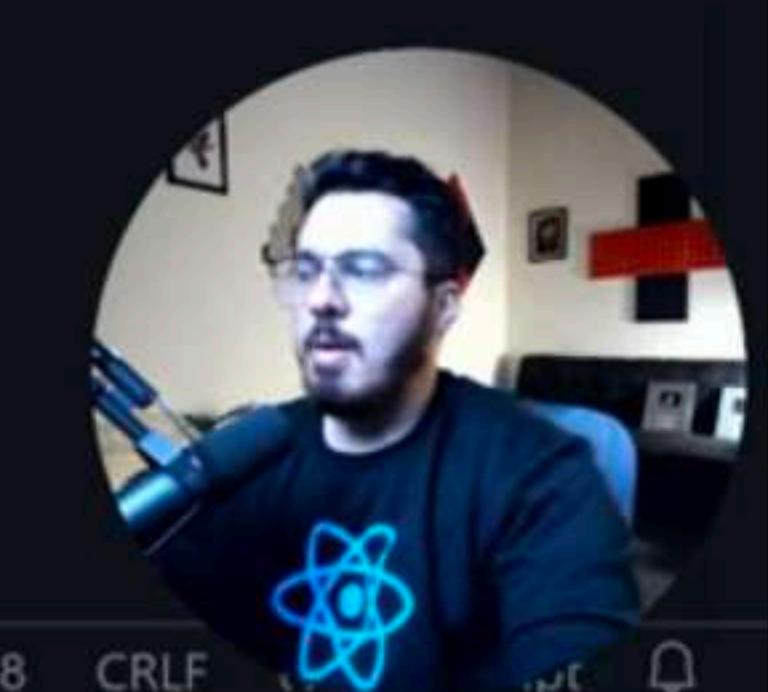
Ln 1, Col 1 Spaces: 4 UTF-8 CRLF

24°C Mostly cloudy

Search

17:45 Wed

Subscribe



File Edit Selection View Go ... ← → Video 120

EXPLORER

VIDEO 120

- > node
- > App.jsx
- JS store.js
- JS main.jsx
- JS counterSlice.js X

src > redux > counter > JS counterSlice.js > counterSlice

```
1 import { createSlice } from '@reduxjs/toolkit'
2
3 const initialState = {
4   value: 0,
5 }
6
7 export const counterSlice = createSlice({
8   name: 'counter',
9   initialState,
10  reducers: {
11    increment: (state) => {
12      // Redux Toolkit allows us to write "mutating" logic in reducers. It
13      // doesn't actually mutate the state because it uses the Immer library,
14      // which detects changes to a "draft state" and produces a brand new
15      // immutable state based off those changes
16      state.value += 1
17    },
18    decrement: (state) => {
19      state.value -= 1
20    },
21    incrementByAmount: (state, action) => {
22      state.value += action.payload
23    },
24  },
25 })
26
```

Ln 7, Col 42 Spaces: 4 UTF-8 CRLF

24°C Mostly cloudy

Search

Subscribe

Sigma Web Development Course - VI Quick Start | Redux Toolkit Vite + React redux-toolkit.js.org/tutorials/quick-start Sigma Web Developme... GitHub - CodeWithHarr... Learn to code online - C... Es with Express

Redux Toolkit

Getting Started Tutorials Usage Guide API RTK Query GitHub

Introduction

Getting Started

Why Redux Toolkit is How To Use Redux Today

Tutorials

Tutorials Overview

Quick Start

TypeScript Quick Start

RTK Query Quick Start

Using Redux Toolkit

Migrations

Usage Guide

Usage With TypeScript

Writing Reducers with Immer

Setup with Next.js

API Reference

RTK Query

RTK Query Overview

Comparison with Other Tools

Examples

Usage With TypeScript

Using RTK Query

API Reference

Add Slice Reducers to the Store

Next, we need to import the reducer function from the counter slice and add it to our store. By defining a field inside the `reducer` parameter, we tell the store to use this slice reducer function to handle all updates to that state.

TypeScript [JavaScript](#)

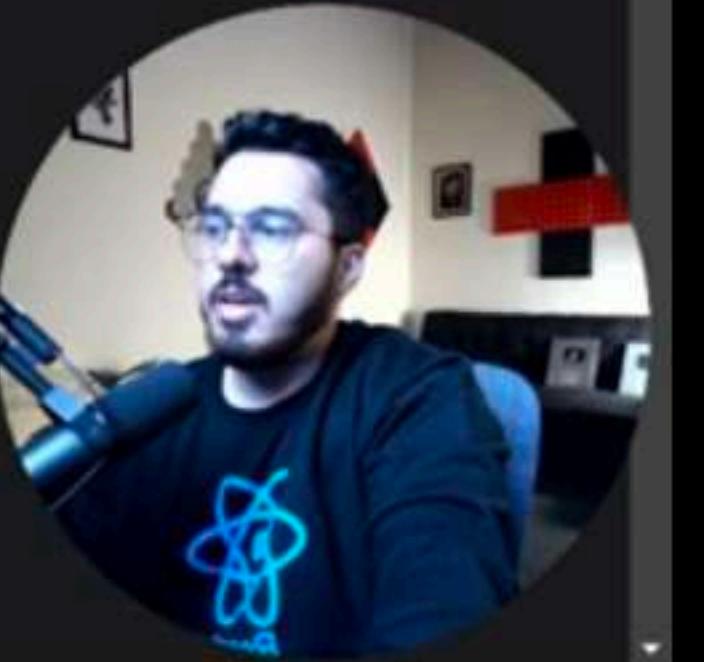
```
app/store.js
import { configureStore } from '@reduxjs/toolkit'
import counterReducer from '../features/counter/counterSlice'

export const store = configureStore({
  reducer: {
    counter: counterReducer,
  },
})
```

Use Redux State and Actions in React Components

Now we can use the React-Redux hooks to let React components interact with the Redux store. We can read data from the store with `useSelector`, and dispatch actions using `useDispatch`. Create a `src/features/counter/Counter.js` file with a `<Counter>` component inside, then import that component into `App.js` and render it inside of `<App>`.

TypeScript [JavaScript](#)



24°C Mostly cloudy

Search

ENG IN 17:46 Wed

Subscribe

File Edit Selection View Go ... ← → Video 120 □ □ □ 00 - o x

EXPLORER node App.jsx JS store.js X main.jsx JS counterSlice.js

VIDEO 120

src > redux > JS store.js > [store]

```
1 import { configureStore } from '@reduxjs/toolkit'
2 import counterReducer from './counter/counterSlice'
3
4 export const store = configureStore({
5   reducer: {
6     counter: counterReducer,
7   },
8 })
```

App.css App.jsx index.css main.jsx .eslintrc.cjs .gitignore index.html package-lock.json package.json README.md vite.config.js

OUTLINE

Timeline

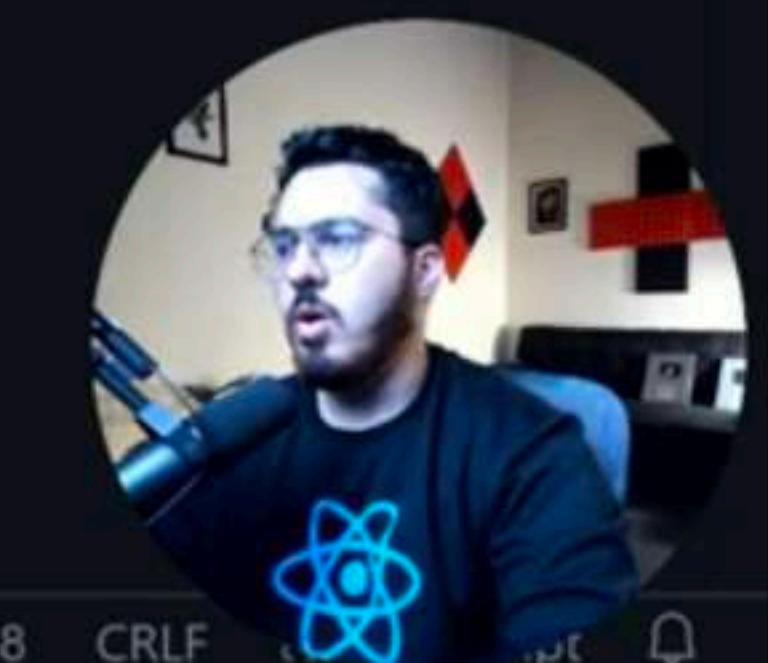
Task Explorer

Ln 7, Col 9 Spaces: 4 UTF-8 CRLF ENG IN 17:48 Wed

24°C Mostly cloudy

Search

Subscribe



Sigma Web Development Course - V... Quick Start | Redux Toolkit Vite + React redux-toolkit.js.org/tutorials/quick-start Sigma Web Developme... GitHub - CodeWithHarr... Learn to code online - C... Ejs with Express

Redux Toolkit

Introduction

Getting Started

Why Redux Toolkit is How To Use Redux Today

Tutorials

- Tutorials Overview
- Quick Start
- TypeScript Quick Start
- RTK Query Quick Start

Using Redux Toolkit

- Migrations
- Usage Guide
- Usage With TypeScript
- Writing Reducers with Immer
- Setup with Next.js

API Reference

RTK Query

- RTK Query Overview
- Comparison with Other Tools
- Examples
- Usage With TypeScript
- Using RTK Query
- API Reference

Use Redux State and Actions in React Components

Now we can use the React-Redux hooks to let React components interact with the Redux store. We can read data from the store with `useSelector`, and dispatch actions using `useDispatch`. Create a `src/features/counter/Counter.js` file with a `<Counter>` component inside, then import that component into `App.js` and render it inside of `<App>`.

TypeScript JavaScript

```
features/counter/Counter.js
```

```
import React from 'react'
import { useSelector, useDispatch } from 'react-redux'
import { decrement, increment } from './counterSlice'

export function Counter() {
  const count = useSelector((state) => state.counter.value)
  const dispatch = useDispatch()

  return (
    <div>
      <div>
        <button
          aria-label="Increment value"
          onClick={() => dispatch(increment())}>
          Increment
        </button>
        <span>{count}</span>
        <button
          aria-label="Decrement value"
          onClick={() => dispatch(decrement())}>
          Decrement
        </button>
      </div>
    </div>
  )
}
```

Introduction

How to Read This Tutorial

Usage Summary

Install Redux Toolkit and React-Redux

Create a Redux Store

Provide the Redux Store to React

Create a Redux State Slice

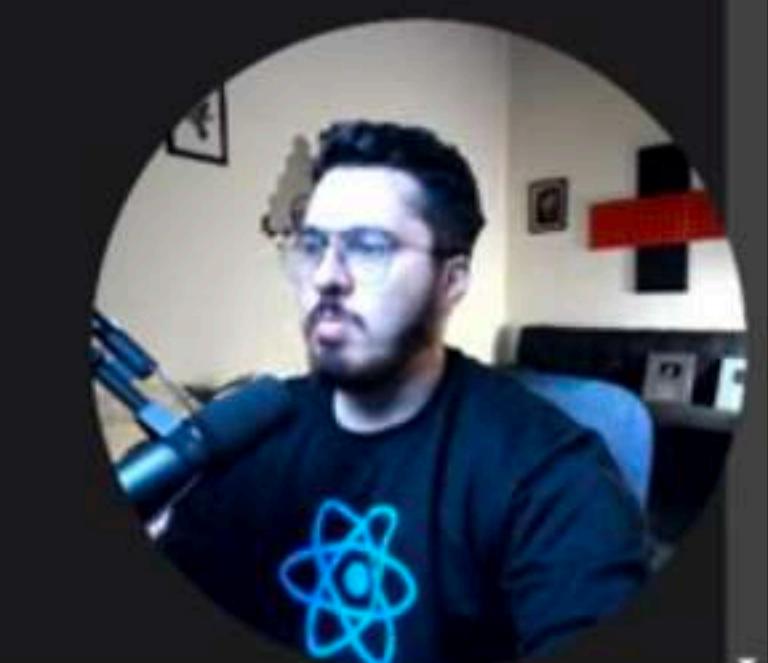
Add Slice Reducers to the Store

Use Redux State and Actions in React Components

What You've Learned

Full Counter App Example

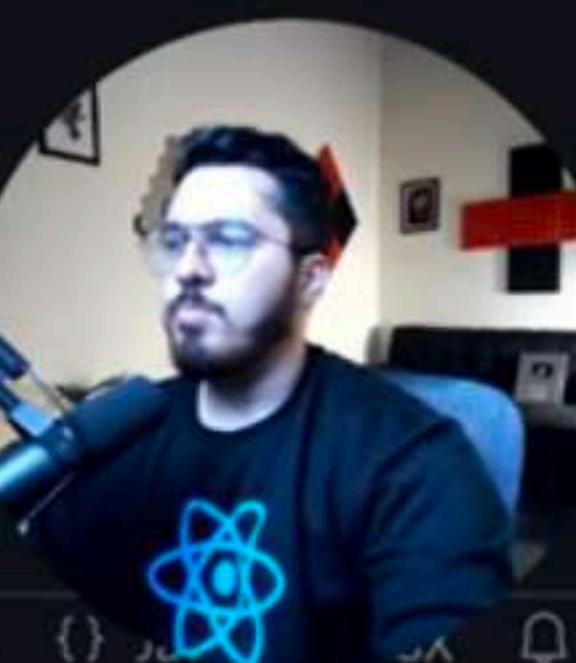
What's Next?



24°C Mostly cloudy

Q Search

ENG IN 17:50 Wed



```
src > App.jsx > App > count
  1 import { useState } from 'react'
  2 import reactLogo from './assets/react.svg'
  3 import viteLogo from '/vite.svg'
  4 import './App.css'
  5 import Navbar from './components/Navbar'
  6 import { useSelector, useDispatch } from 'react-redux'
  7
  8 function App() {
  9   const [count, setCount] = useState(0)
 10   const count = useSelector((state) => state.counter.value)
 11
 12   return (
 13     <>
 14       <Navbar />
 15       <div>
 16         {count}
 17       </div>
 18
 19     </>
 20   )
 21 }
 22
 23 export default App
 24
```

File Edit Selection View Go ... ← → Video 120

EXPLORER

VIDEO 120

- > node_modules
- > public
- src
 - > assets
 - > components
 - redux
 - counter
 - JS counterSlice.js
 - JS store.js
 - App.css
 - App.jsx
 - index.css
 - main.jsx
- .eslintrc.cjs
- .gitignore
- index.html
- package-lock.json
- package.json
- README.md
- vite.config.js

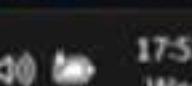
OUTLINE

TIMELINE

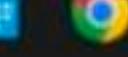
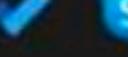
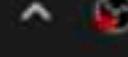
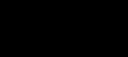
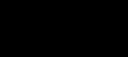
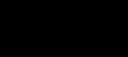
TASK EXPLORER

node App.jsx X store.js main.jsx counterSlice.js

```
src > App.jsx > App
1 import { useState } from 'react'
2 import reactLogo from './assets/react.svg'
3 import viteLogo from '/vite.svg'
4 import './App.css'
5 import Navbar from './components/Navbar'
6 import { useSelector, useDispatch } from 'react-redux'
7
8 function App() {
9   const count = useSelector((state) => state.counter.value)
10
11   return (
12     <>
13       <Navbar />
14       <div>
15         {count}
16       </div>
17     </>
18   )
19 }
20
21
22 export default App
23
```

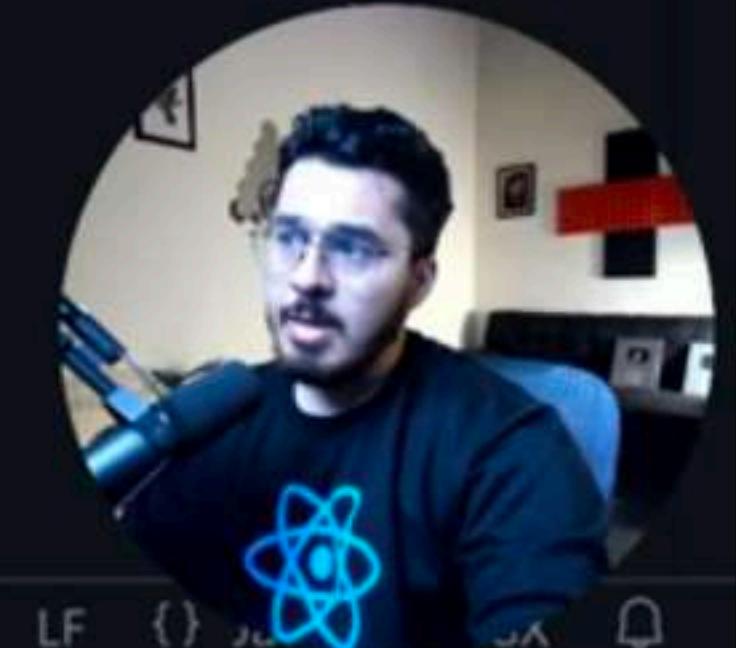
Ln 10, Col 1 (52 selected) Spaces: 2 UTF-8 LF {}  

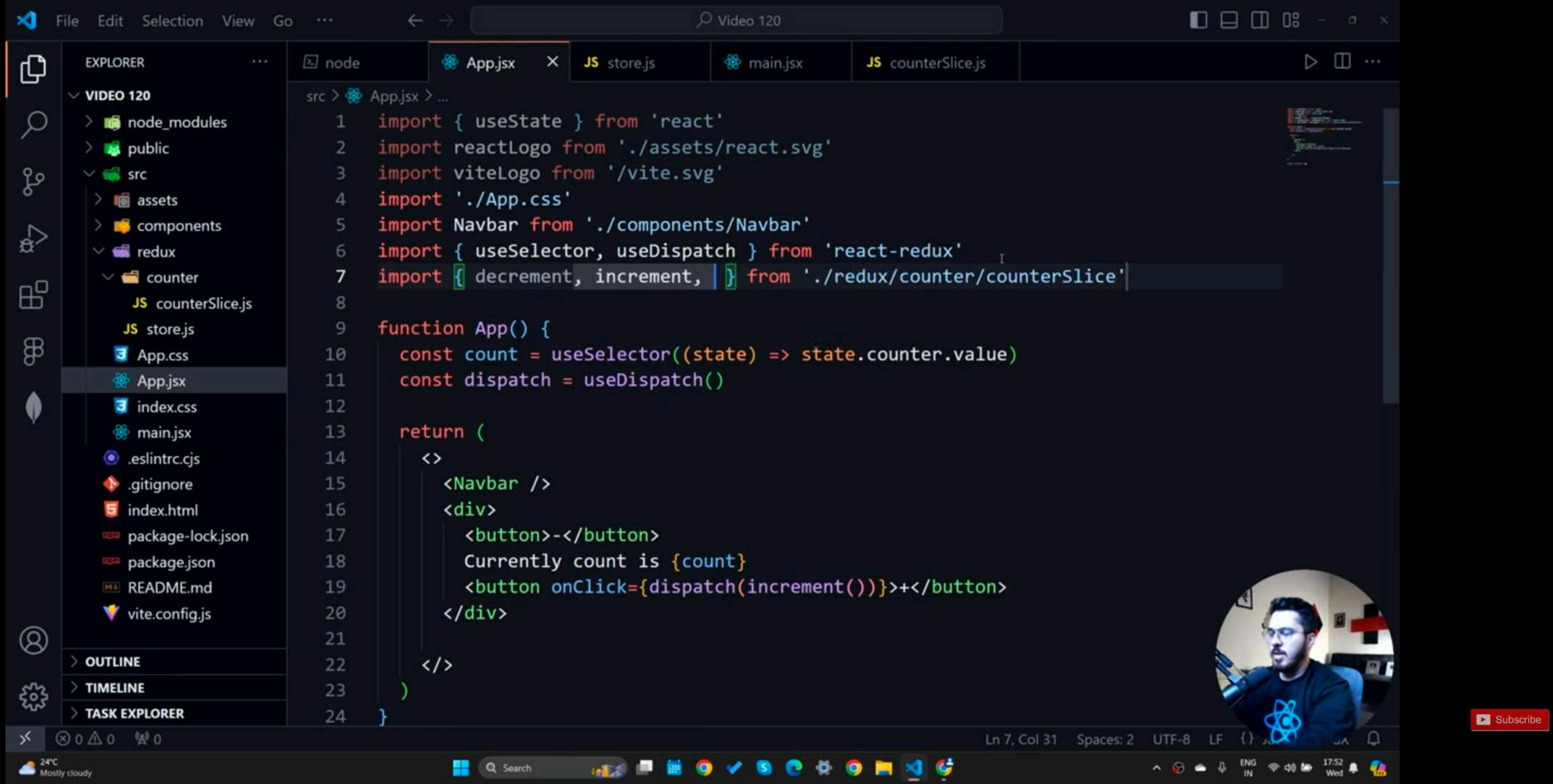
24°C Mostly cloudy

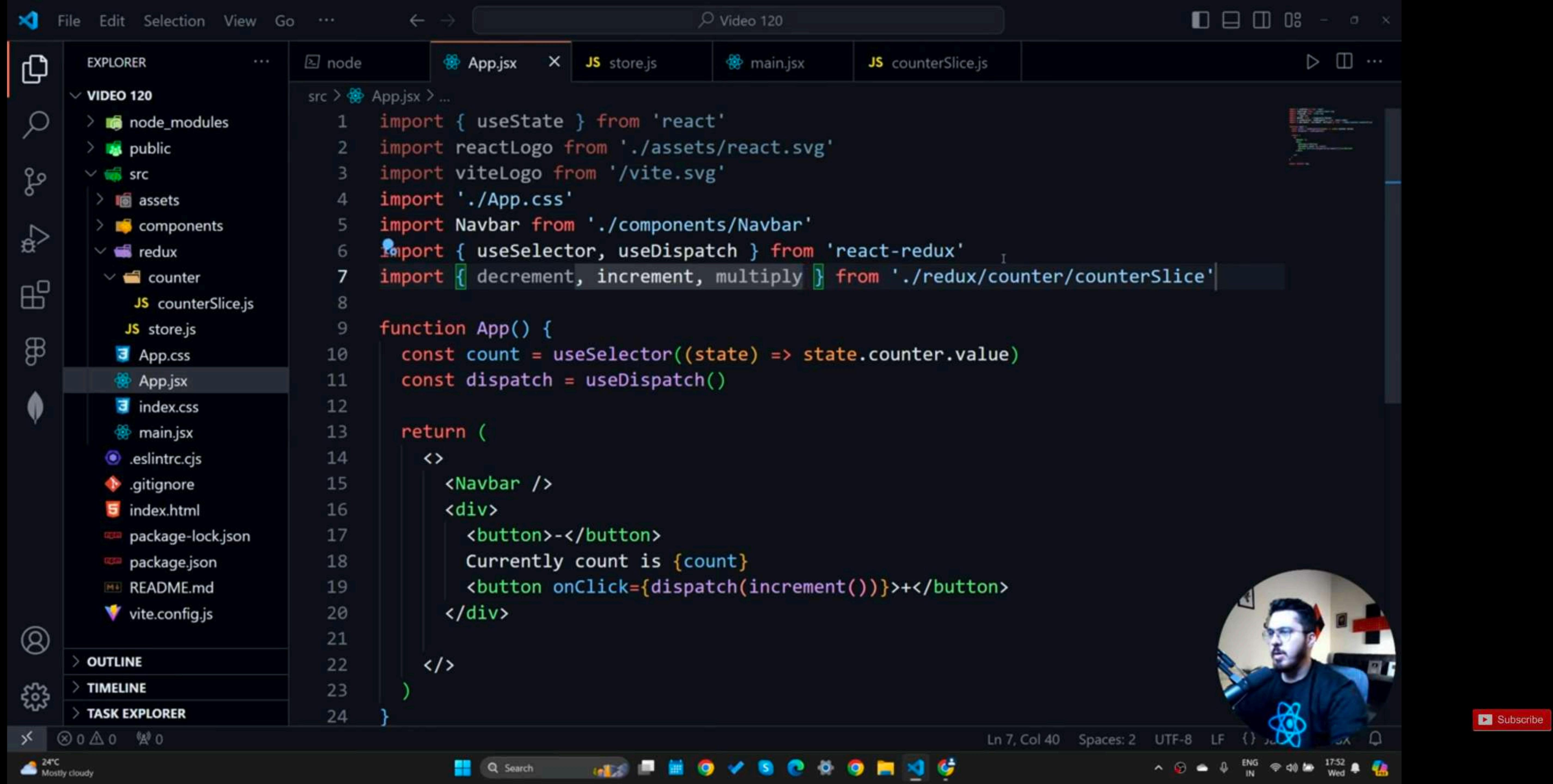
Search                      

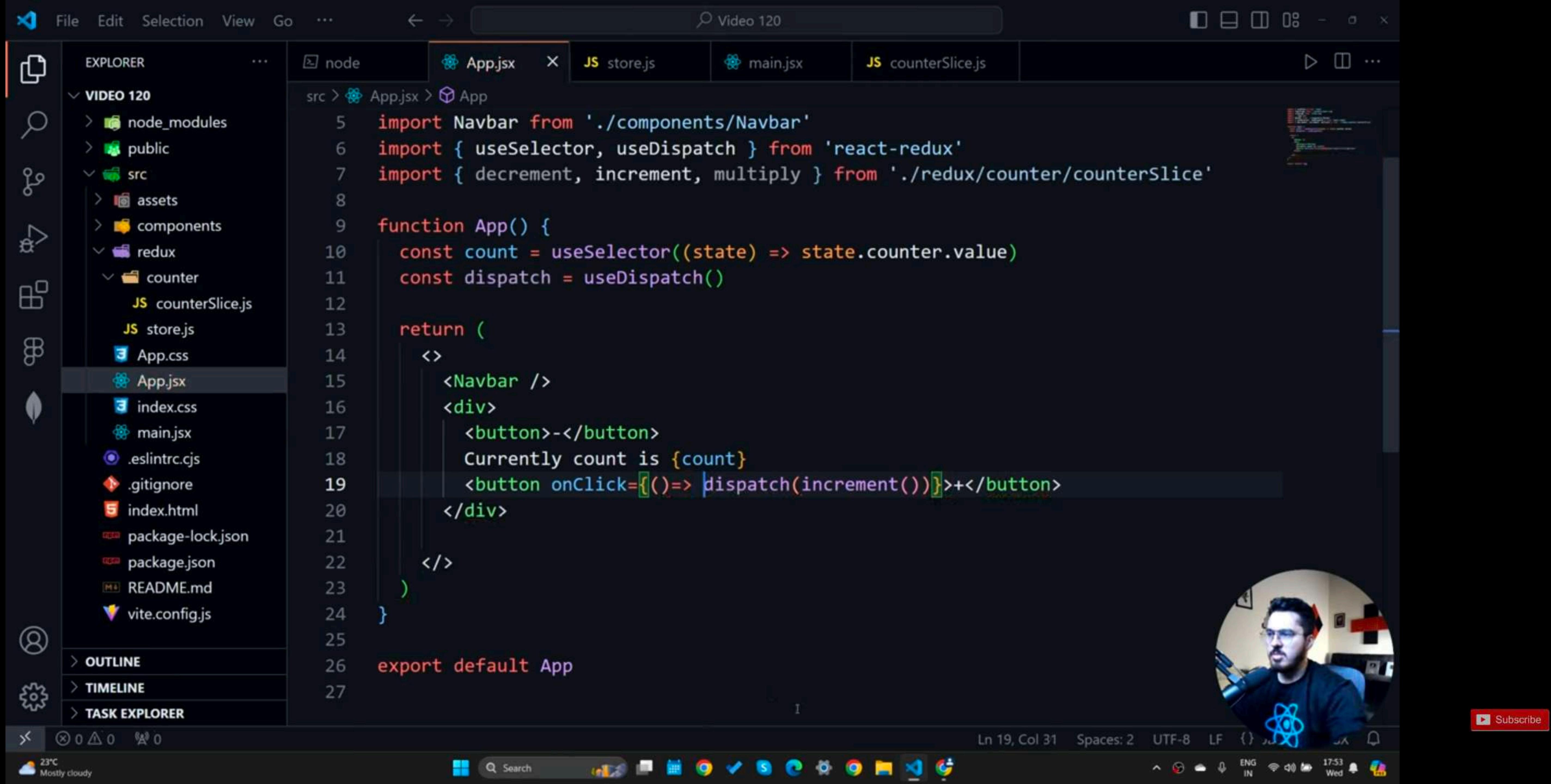
ENG IN 17:51 Wed

Subscribe









File Edit Selection View Go ... ← → Video 120

EXPLORER

VIDEO 120

- > node
- > App.jsx
- > JS store.js
- > main.jsx
- > JS counterSlice.js X

src > redux > counter > JS counterSlice.js > multiply

```
7  export const counterSlice = createSlice({
10    reducers: {
11      increment: (state) => {
12        state.value += 1
13      },
14      decrement: (state) => {
15        state.value -= 1
16      },
17      incrementByAmount: (state, action) => {
18        state.value += action.payload
19      },
20      multiply: (state) => {
21        state.value *= 2
22      }
23    },
24  })
25  // Action creators are generated for each case reducer function
26  export const { increment, decrement, incrementByAmount, multiply } = counterSlice.
27  actions
28
29
30
31
32
33  export default counterSlice.reducer
```

OUTLINE

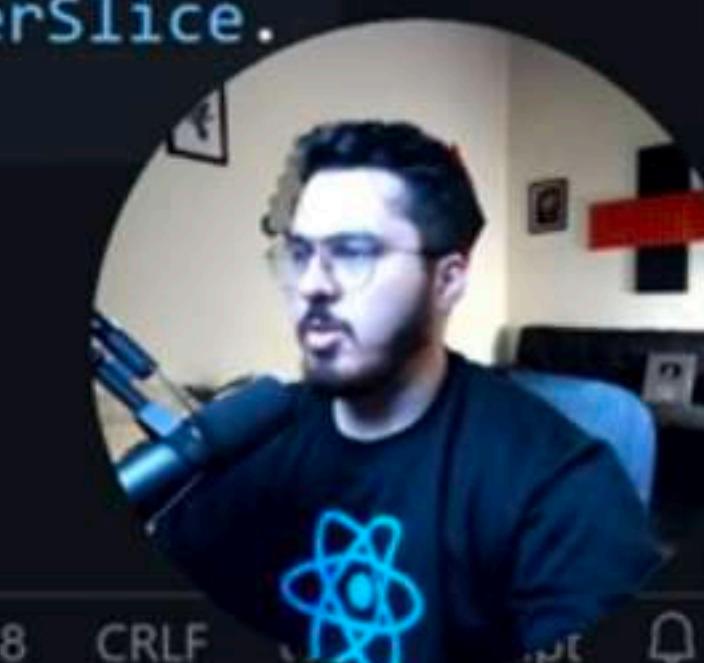
TIMELINE

TASK EXPLORER

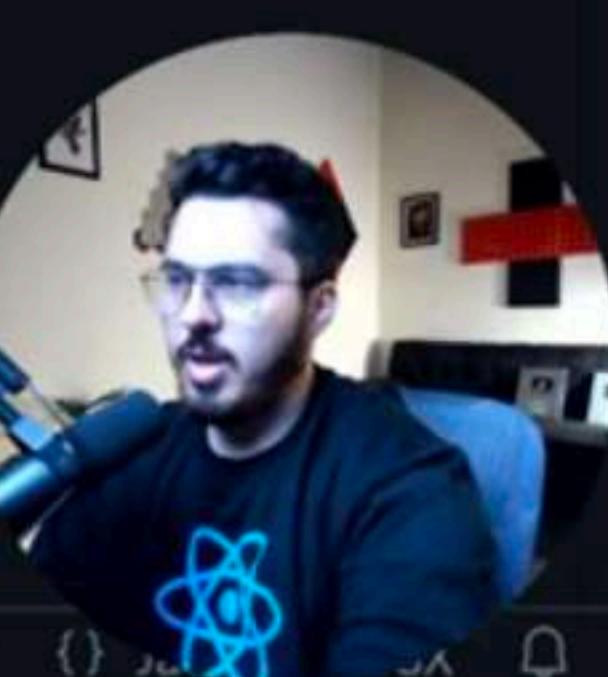
Ln 31, Col 65 Spaces: 4 UTF-8 CRLF ENG IN 17:54 Wed

23°C Mostly cloudy

Search



Subscribe



```
src > components > Navbar.jsx > ...
1 import React from 'react'
2 import { useSelector, useDispatch } from 'react-redux'
3
4 const Navbar = ({count}) => {
5   return (
6     <div>
7       I am a navbar and counter is {count}
8     </div>
9   )
10 }
11
12 export default Navbar
13
```

File Edit Selection View Go ... ← → Video 120

EXPLORER

VIDEO 120

- > node_modules
- > public
- < src
 - > assets
 - < components
 - Navbar.jsx

redux

- < counter
 - JS counterSlice.js
 - JS store.js
 - App.css
 - App.jsx
 - index.css
 - main.jsx

.eslintrc.cjs

.gitignore

index.html

package-lock.json

package.json

README.md

vite.config.js

OUTLINE

TIMELINE

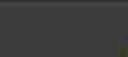
TASK EXPLORER

node App.jsx store.js Navbar.jsx X main.jsx counterSlice.js

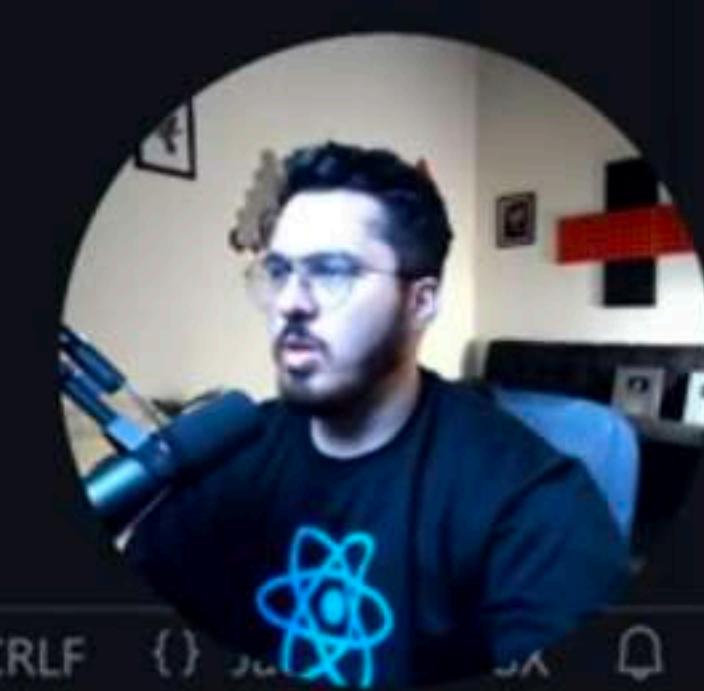
```
src > components > Navbar.jsx > Navbar
1 import React from 'react'
2 import { useSelector, useDispatch } from 'react-redux'
3
4 const Navbar = ({count}) => {
5   const count = useSelector((state) => state.counter.value)
6   return (
7     <div>
8       I am a navbar and counter is {count}
9     </div>
10  )
11 }
12
13 export default Navbar
14
```

Ln 5, Col 60 Spaces: 2 UTF-8 CRLF { }  

23°C Mostly cloudy

Search         

17:56 Wed ENG IN  



Subscribe

File Edit Selection View Go ... ← → Video 120

EXPLORER

VIDEO 120

- > node_modules
- > public
- < src
 - > assets
 - < components
 - Navbar.jsx

redux

- < counter
 - JS counterSlice.js
 - JS store.js
 - App.css
 - App.jsx
 - index.css
 - main.jsx

.eslintrc.cjs

.gitignore

index.html

package-lock.json

package.json

README.md

vite.config.js

OUTLINE

TIMELINE

TASK EXPLORER

node App.jsx store.js Navbar.jsx X main.jsx counterSlice.js

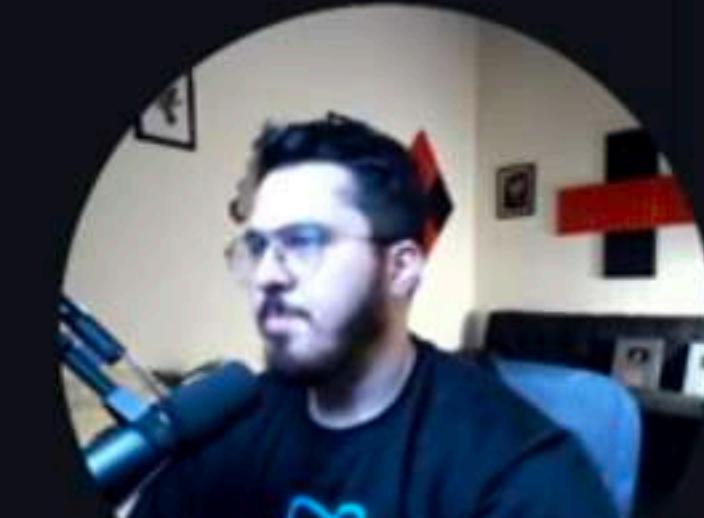
```
src > components > Navbar.jsx > Navbar
1 import React from 'react'
2 import { useSelector, useDispatch } from 'react-redux'
3
4 const Navbar = () => {
5   const count = useSelector((state) => state.counter.value)
6
7   return (
8     <div>
9       I am a navbar and counter is {count}
10    </div>
11  )
12}
13
14 export default Navbar
15
```

Ln 4, Col 17 Spaces: 2 UTF-8 CRLF { }  

23°C Mostly cloudy

Search         

ENG IN 17:56 Wed

File Edit Selection View Go ... ← → Video 120

EXPLORER

VIDEO 120

- > node_modules
- > public
- > src
 - > assets
 - > components
 - Navbar.jsx
 - > redux
 - > counter
 - counterSlice.js
 - store.js
 - App.css
 - App.jsx
 - index.css
 - main.jsx
 - .eslintrc.cjs
 - .gitignore
 - index.html
 - package-lock.json
 - package.json
 - README.md
 - vite.config.js

node App.jsx store.js Navbar.jsx main.jsx counterSlice.js

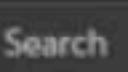
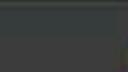
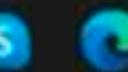
```
src > App.jsx > App
4 import './App.css'
5 import Navbar from './components/Navbar'
6 import { useSelector, useDispatch } from 'react-redux'
7 import { decrement, increment, incrementByAmount, multiply } from './redux/counter/
counterSlice'

9 function App() {
10   const count = useSelector((state) => state.counter.value)
11   const dispatch = useDispatch()

13   return (
14     <>
15       <Navbar />
16       <div>
17         <button onClick={() => dispatch(decrement())}>-</button>
18         Currently count is {count}
19         <button onClick={() => dispatch(incrementByAmount(1))}>+</button>
20       </div>
21     </>
23   )
24 }
25
26 export default App
```

Ln 19, Col 59 Spaces: 2 UTF-8 LF {} 

23°C Mostly cloudy

Search        

Subscribe



